COMPUTER MODELLING OF INFORMATION THAT IS CONTINUOUS OVER A SPATIAL DOMAIN

A THESIS
SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE IN COMPUTER SCIENCE
IN THE
UNIVERSITY OF CANTERBURY
BY

DEAN GRAEME ASHBY

University of Canterbury
1995

Abstract

Fitting data that vary continuously over an area of land into a discrete data model can introduce a high level of error. The work done in this thesis deals with this problem by exploring the use of alternative data structures and processing methods to represent better those features of the environment being analysed. The fuzzy c-means (FCM) classification algorithm has been used to measure the variation of geographic features over a spatial domain, and to output this information in the form of a membership raster for each feature. A membership raster is similar to a raster-based digital elevation model (DEM). Therfore a triangular irregular network (TIN) can also be used to represent a membership raster. A TIN representation requires less storage space than a membership raster, and allows the algorithms that have been developed for processing terrain information to be used for processing membership values.

Acknowledgements

I wish to express my appreciation for the valuable guidance offered by Professor John Penny during the preparation of this thesis, and to Dr Pip Forer who offered insightful comments throughout.

I also wish to thank Richard Pascoe for his support, from someone who has been there, and done that.

I would like to acknowledge Mike Hayton for his assistance in providing program code to implement the algorithm used to add multiple TINs together.

To all my fellow postgraduate researchers, thank you for your companionship over the last two years. I hope the names and faces will never fade.

Finally, to my parents, who have unwaveringly supported me throughout my university career. Thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis is concerned with the computer modelling of information that is continuous over a spatial domain. Fitting continuous data into a discrete data model can introduce a high level of error. For example, the segmentation and classification of land areas into regions with sharp boundaries can lead to an inaccurate representation of the real environment. In the process of classifying continuous data into discrete groups or classes, useful information is lost, such as the way in which the data vary over the spatial domain.

Taxonomic processes aim to classify areas or objects into distinct classes. Various statistical methods are used to determine to which class an object is most likely to belong, based on some characteristics of the object. A measure of accuracy is often also recorded, to describe the probability that an object has been classified correctly.

A further extension is to calculate and record the probability of an object belonging to each class. The fuzzy c-means (FCM) algorithm (Bezdek, Ehrlich & Full 1984) is a classification algorithm that uses fuzzy logic to calculate the similarity of an object to each class. Similarity is represented by a *membership function* whose values or *memberships* range from zero to one, each object having a degree of membership in each class. The algorithm can be used to calculate the degree of membership to each class for each pixel of a remote sensed image or any other raster data. The FCM algorithm classifies one object at a time, ignoring neighbouring objects. Thus the algorithm can be applied to irregularly spaced objects.

Classification into regions or "spatial discretisation" of data still presents a problem when modelling continuous change over space. Results from the FCM algorithm can be used to assist in the interpolation of values for intermediate locations. As a result of classifying a collection of pixels using the FCM algorithm, a large amount of data is produced, thereby causing storage problems. Information theory and context modelling

can be used to identify *critical points* in a data set, thus reducing the number of points without losing information. Critical points are those which are unusual or different from other points in a set. Context modelling refers to comparing a point to its neighbours to determine if it is different, rather than comparing it to the entire set of points.

Once critical points have been identified, other non-critical or redundant points can be discarded as they are likely to contain little additional information. Because critical points are distributed irregularly in two dimensions, a triangular irregular network (TIN) can be used as a storage structure. The TIN is used to represent levels of class membership, rather than physical elevation as in a terrain model. By using a TIN to represent membership levels, the spatial distribution of data points can be altered so that a few points are used to represent areas of gradual change, and many points are used to represent areas where membership values change sharply. Using an on-line triangulation algorithm, it is possible to form the union, or intersection, of class membership representations for several classes.

The entire process outlined here for modelling continuous change over space requires more information to be gathered than for traditional classification and discretisation techniques. However attempts have been made to limit the volume of data by selective removal of unwanted data, and the use of appropriate data structures to reduce storage requirements. By calculating class membership values for critical points over space, a more realistic model of the environment can be built, in turn allowing more accurate conclusions to be drawn about the spatial variation of phenomena.

This thesis is structured as follows:

In Chapter 2, a description is given of the various types of error that can occur in a GIS. Most errors can be grouped into two types, locational error, and attribute error. Examples are also discussed of methods that have previously been used to reduce or model such errors.

In Chapter 3, classical logic theory is compared to fuzzy logic theory by using set theory to describe the differences. A method for converting fuzzy information into discrete informations is given, followed by a discussion of the relevance of fuzzy logic to the process of classifying pixels.

In Chapter 4, the characteristics of the remotely sensed image used in this research are described. Methods are also given by which fuzzy logic can be used to extract useful information from the satellite imagery. A large proportion of this information is not essential, as general trends in the spatial distribution of features can be observed using a few critical points.

In Chapter 5, four approaches are given for identifying critical points in the data produced by the method described in Chapter 4. By using one of these approaches, the volume of data can be reduced while retaining a high level of information.

In Chapter 6, appropriate data structures are investigated for storing critical points. Due to the irregular spatial distribution of critical points, the TIN data structure was chosen by the author as being the most suitable.

Finally, in Chapter 7, a number of conclusions are drawn: that features that vary continuously over a spatial domain are poorly represented using discrete data structures; that fuzzy classification captures more information allowing a more accurate approximation of continuous features; that the extra information produced by a fuzzy classification necessitates data reduction schemes and the use of more efficient data structures.

The main conclusion is that fuzzy classification is a useful tool for gathering information about the distribution of features that vary continuously over a spatial domain. However, such classification must be accompanied by methods for reducing the volume of data to a manageable size. Finally, the effectiveness of these methods is highly dependent on the amount of variation present in the features.

Chapter 2

Dealing with Error in Geographic Information Systems

Representing some kinds of continuous information in a GIS may be difficult. GISs have been extensively and successfully used for representing and analysing non-naturally occurring features, and this is reflected by the proliferation of facilities management applications for GIS. Naturally occurring features tend to be less precise in their definition, and are thus harder to model in a GIS.

This chapter describes research on some of the problems involved with modelling natural features and, in particular, those features that vary continuously over space.

The first section describes previous research in the field of representing continuous information. As well, error in geographic information systems (GIS), the sources of error, and its characteristics are discussed. Several approaches are described for modelling of the environment and dealing with the errors associated with particular models.

2.1 Types of error

Data quality is an important aspect of maintaining a GIS. There appear to be two major approaches to dealing with error. The first approach has involved standardising meta-data, that is, data that describe data, in an attempt to record the history of data. In the case of geographic information, meta-data usually include information such as: the date of data capture, how the data were collected, and a record of transformations or projections applied to the data. In this way, the history of a data set can be traced from its collection through all subsequent processing steps.

The second approach has been concerned with modelling the error associated with

various operations performed in the process of collecting data and analysing data contained within a GIS. Whenever an operation that alters the attributes or coordinates of a feature is performed, it is possible that errors may be present in the new values. By calculating the potential type and amount of error introduced by each process, a model can be built of the error associated with each operation. Combining this information with a meta database, an estimate of the error associated with a data set can be made.

The approach described by Chrisman (1989) is to categorise error into: locational error, and incorrect attribute values or classification error. Locational error is the difference between the location of an object in reality, and the location actually represented in a GIS. Classification error is the difference between what an object is in reality and what it is defined to be in a GIS.

Errors have been categorised by Goodchild (1989) into two main groups, source error and process error. Source error is the difference between geographical truth and the abstract model that we use to represent it. Process error includes any form of error that is introduced as a result of digital processing of geographic data, including the initial data capture process. Goodchild (1989) describes an object in a spatial database as being an "abstract model of real, continuous and complex geographic variation".

2.1.1 Source error and process error

Common types of error in GISs can be classified into two main groups: errors that result from using models not able to represent every characteristic of the object being modelled, and errors that result from data acquisition and processing. An example of source error is the representing of a stand of trees as one homogeneous block, when there are small areas within the block that are not vegetated. Processing error can be, for example, the result of human error when digitising, or rounding errors when transforming geographic coordinates between different projections.

Representations of artificial features, such as census tract boundaries or building foot-prints, are often free of source error because they can be adequately represented using discrete objects such as arcs and polygons. When representing more complex features, particularly those resulting from some natural process, source error is likely to exceed processing error (Goodchild 1989). Improving the data models used to represent natural features may reduce the difference between geographic reality and the digital representation, and hence reduce the amount of source error.

2.1.2 Continuous and discrete data

Vector-based GISs use discrete objects such as points, lines, and polygons to represent geographic features. Provided that the features represented are also discrete, then the vector-based GIS can provide an appropriate environment for manipulating and processing information about those features.

However, many geographic features are not discrete. Features that are the result of natural processes acting over long periods of time usually do not have sharp boundaries. To represent these features using discrete objects, the boundaries of the features must be approximated. The approximation process may be performed manually by a cartographer, or automatically using a computer.

Locational error and classification error are very closely related when modelling continuous information. Locational error is the difference between the true geographic location of a feature and the location as represented in a GIS. The location of a boundary between regions often depends on how the regions are classified. Classification error results from taking a continuous variable and approximating it with a discrete value or boundary. Typically, natural features are transformed from continuous objects to discrete objects in two ways. First, the gradual change at their boundaries is approximated by a line. Second, the area within the polygon formed by the boundary lines is assumed to be homogeneous.

2.2 Error models

There have been a number of approaches to handling locational and classification error in geographic data. These methods range from simulating natural variation using fractal functions, to calculating concentric error bounds around features by using fixed tolerance levels. Fuzzy logic has been used in a limited way to reduce some of the error associated with Boolean classification. The following sections discuss several of these models, and how the ideas behind the models have produced better methods for representing continuous information.

2.2.1 Fractal models

Research by Mandelbrot (1977, 1983) in the field of statistics resulted in functions that could produce realistic and striking simulations of landscape. Mandelbrot's functions are known more widely as fractals. The term fractal is derived from the term *fractional dimension*. The fractional dimension D can be thought of as a measure of the variation of

a line. A straight line has a value of 1.0 for D. A line that changes direction so often that it completely covers a given area, has a value of 2.0 for D. The value of D for a given line can be used to classify the line in terms of its uniformity.

A second characteristic of fractals is the property of self-similarity. Curves and lines that are said to be self-similar have a constant value for D, independent of the scale at which D is calculated. A geographic example of self-similar lines is the set of lines representing a river bead at various scales. The pattern of spatial variation in the river bed at one scale is repeated at a completely different scale.

These two characteristics of fractals are interesting as it was initially thought that they could be used to produce simulated variation along digitised lines. If the amount of variation at the boundary of a feature can be stated using D, then this can be used to produce approximations of the same line at different scales. The concept of self-similarity is not limited to lines, and can also be applied to simulating the distribution of small island polygons at the edge of larger polygons. It was shown by Goodchild (1980) and Burrough (1984) in this context, that the self-similarity property did not occur often in lines that represented natural features.

The reason is that different scales show the effects of different processes. Consider, for example, a coastline. The shape of an island, when shown at a small scale, may be the result of forces involved with plate tectonics. At a larger scale will be shown the weathering effects of waves that determine the shape of bays and headlands. At an even larger scale, we see the shape of individual rocks as influenced by chemical reactions with the atmosphere and sea water. All these processes produce different patterns, so the lines representing a coastline at different scales are not likely to be self-similar.

2.2.2 Tolerance models

A common feature of spatial operations is the automatic joining or *snapping* of features within a given range of each other. For example, when the union of two polygons is formed, two lines representing part of each polygon's boundary may be very close together. If the lines are within a given snapping distance, one of the lines may be chosen, and the other removed from the set of lines representing the union. The snapping distance is usually calculated from the scale and coordinate units associated with the data, but not necessarily according to the type of feature.

Maffini, Arno & Bitterlich (1989) suggested the idea of concentric error bounds. Based on some snapping distance, concentric error bounds can be defined that radiate from

a geographic feature. In the case of a boundary between two polygons, the line representing the boundary may be considered the place of maximum uncertainty in terms of membership of one polygon or the other. This approach can be reversed by arguing that the location of the line on a map is the position of a feature, given the errors in data capture and processing. As the distance away from the line increases, the probability that the new location could be the real position of the feature decreases.

Maffini et al. (1989) applied these ideas, and defined probability distributions for the locational error associated with particular geographic features. The distribution closely resembled a normal distribution, with the most likely position of the feature being located at the mean. The distributions associated with well delineated features such as roads have a very low standard deviation, with probabilities decreasing rapidly as distance from the feature is increased. The boundaries of a drainage basin are harder to define, so the standard deviation for the distribution is increased relative to the former case. Therefore, probability decreases very slowly as the distance from the mean increases.

By taking these probability distributions into account when forming the intersection of points, lines and polygons during a spatial overlay operation, a map can be produced showing the intersection of the probability distributions around each of these features. Maffini et al. (1989) used, as an example, a spatial query based on three criteria: areas must be within 100 meters of a road, have a slope greater than 15%, and be located within a particular drainage basin. The lines and polygons representing each of the features specified in the query have an associated probability distribution. Instead of forming the intersection of these features, the probability of an area fulfilling each of the criteria in the query is calculated, and the combination of probabilities for all criteria is formed. The resulting map contains a collection of grey shaded areas. The darker the shade of grey, the more likely the area is to satisfy the criteria given in the query.

2.2.3 Fuzzy models

The models described so far have dealt only with locational error due to processing and representation methods. Fuzzy models have been used in an attempt to reduce the amount of source error resulting from using Boolean classification methods for analysis of land suitability. Chapter 3 describes in detail the differences between fuzzy and Boolean classification.

Wang, Hall & Subaryono (1990) designed and tested a fuzzy classification scheme for use in land suitability analysis. Further to this, Hall, Wang & Subaryono (1991) compared the results of a Boolean classification and a fuzzy classification, concluding that the

advantages of the latter method far outweighed those of the Boolean classification.

When using Boolean classification to determine land use suitability, several characteristics, such as temperature, rooting conditions, nutrient availability, toxicity, and terrain, may be measured over a study area. Characteristics such as these are chosen because they influence what could be the most economic use of an area of land.

For each characteristic, the study area is subdivided into smaller areas of either high values or low values of that characteristic. For example, suppose that nutrient levels are measured across the study area, and the study area is then divided into a series of polygons. Each polygon has either high levels or low levels of nutrients available in the soil. Similarly, for terrain the study area might be divided into areas of slope less than 15%, and areas of slope $\geq$ 15%. Rules can be defined for each type of land use by using Boolean decisions based on each characteristic. If an area conforms to the rule for a particular use, then the area is suitable for that use. If the area breaks the rule, then it is not suitable for that use.

By using a fuzzy classification derived from pattern recognition, the range of levels for each characteristic can be expanded. Intermediate levels can be defined for each characteristic. Hall et al. (1991), defined four mutually exclusive levels, ranging from unsuitable to highly suitable. Areas were then classified to one of the possible four levels. With respect to source error, the model described here provides a more accurate representation of the real situation than the representation provided by the Boolean model.

Other work using fuzzy models includes fuzzy representation of boundaries. Thematic maps, as implemented in many GIS, are usually composed of a collection of points, lines, and polygons. The contents of polygons are often assumed to be homogeneous, with sharp boundaries. In certain cases this assumption is valid, for example for the property boundaries in a cadastral database. For other types of data, however, particularly for themes that are the result of some natural process, these assumptions are not valid (Leung 1987, Mark & Csillag 1989, Wang & Hall 1991).

Wang & Hall (1991) describe a method for calculating and representing the fuzziness of a boundary. A boundary's fuzziness is obtained by calculating the amount of change between the two adjacent polygons. Boundaries separating polygons with properties of significant difference are termed non-fuzzy, or *crisp*, whereas boundaries between polygons of similar value are more difficult to locate, and are therefore considered fuzzy. For areas with multiple attribute variables, the fuzziness is calculated by calculating the Euclidean distance of the areas in n-dimensional space, where n is the number of attribute variables recorded for each area.

2.3 Summary

A description of the types of error that can result from representing geographic features in a digital form has been given. Errors can be divided into two classes, source error, and processing error. These two terms refer to how the error was introduced: either because of using data models that cannot accurately represent geographic features; or as the result of some process that is applied to the data during, for example, line digitising or polygon overlay.

Errors that occur in digital data can also be divided into two other classes, based not on how the error was introduced, but on how error affects the data. Locational error is the difference between the true location of a feature, and the location of the object as specified in a GIS. Classification error is concerned not with the physical position of an object, but with the attribute values associated with an object. Classification error is related to source error. If a data model is not able to represent all the possible values for an attribute of an object, then classification error will occur.

Several methods designed to avoid some of the error problems have been described. These methods provide ideas relevant to the problems of representing features that are present over a continuous area, and whose values vary continuously over that area. These ideas include the use of probability distributions for modelling locational error, and the application of fuzzy classification to represent attribute values that vary continuously. The role of fuzzy logic is important, as it allows a much greater amount of information to be used for later processing. The next chapter will discuss fuzzy logic and fuzzy classification in more detail, and provide comparisons to the use of Boolean logic and traditional image classification schemes.

Chapter 3

An Introduction to Fuzzy Logic

The previous chapter discussed how fuzzy logic has been partially applied to solve the problem of classification error by increasing the class membership range from two values to four values, ideally the range should be an infinite number of values between 0 and 1. This chapter describes the differences between classical logic and fuzzy logic. A method for converting fuzzy information to discrete information is given in Section 3.1.2, where the alpha cut operation is described. The chapter ends with a brief discussion of the significance of fuzzy logic for the process of classifying pixels within a remote-sensed image.

3.1 Fuzzy sets

The difference between fuzzy logic and Boolean logic is best explained by a comparison of fuzzy set theory and classical set theory. Classical set theory states that an object, x for example, is either a member of some set A, or is not a member of set A. The two conditions are mutually exclusive, and are represented in the following manner:

$$x \in A$$

or

$$x \notin A$$

The process by which objects are determined to be either members or non-members of a set can be defined by a characteristic function. A characteristic function defines some criterion that all members of a set must have. For a given set A, the characteristic function produces a value $\mu_A(x)$ for every element x, such that:

$$\mu_A(x) = \begin{cases} 1 \text{ if and only if } x \in A, \\ 0 \text{ if and only if } x \notin A. \end{cases}$$

This characteristic function maps elements from the universal set to the set containing 0 and 1. The mapping of elements is represented by:

$$\mu_A : X \rightarrow \{0, 1\}$$

The sets that result from mapping elements to 0 or 1 can be termed *crisp* sets. Boolean logic is based on classical set theory, with the values 0 and 1 equivalent to *true* and *false*. All Boolean decisions can produce only the discrete values of true and false as a result. Boolean logic and classical set theory cannot therefore be used to represent elements that may have partial membership in a set. When an element has some but not all of the properties associated with the members of a set, the element is said to have partial membership in the set. Zadeh (1965) was addressing cases such as this when he proposed the theory of fuzzy sets (Klir & Folger 1988).

Fuzzy set theory is different from classical set theory in that the characterisation function has been generalised so that the values produced by the function are continuous, falling within a specified range, and indicate the grade of membership of an element in a given set. The generalised characterisation function is referred to as a membership function, as it maps elements to a set that contains all possible levels of membership in another set. A high value for the membership function indicates a high level of membership for the element in the set associated with the membership function. Similarly, a low value for the function means a low level of membership for the element in the set. The mapping of elements from the universal set X, to the set containing all possible membership values for the fuzzy set A, may take the form:

$$\mu_A : X \rightarrow [0, 1]$$

This mapping indicates that membership values can be any real number in the range from 0 to 1. Membership functions are usually more complex than characterisation functions, as they can produce an infinite number of values, as compared to characterisation functions that produce only two values. As an example, the set of numbers that may be considered the real numbers close to zero could be defined by the following membership function:

$$\mu_A(x) = \frac{1}{1 + 10x^2} \tag{3.1}$$

This membership function can be plotted on a graph as shown in Figure 3.1. The y value of the curve represents the the membership level of the number x in the set of real numbers close to zero. As x approaches zero, the membership level increases, and then decreases as x moves further away from zero.

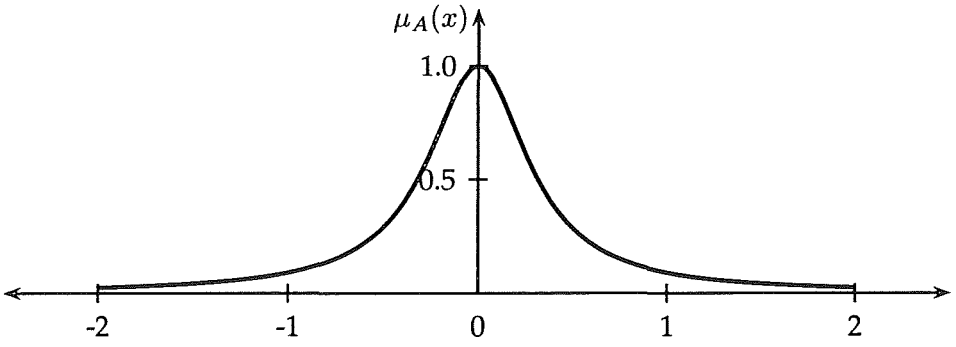


Figure 3.1: Membership function for real numbers near zero.

Membership levels should not be mistaken for a measure of probability. In some cases, several fuzzy sets may be defined, and an element may have a level of membership in each fuzzy set. When an element's membership level in each set are summed, the result may be a value of 1. However, this is not a requirement of membership functions that are defined on the same variable. A probability value describes the likelihood that one of several possible events will occur, and the events are mutually exclusive. Membership values differ because fuzzy sets are not mutually exclusive so the sum of membership values may be greater than 1.

A further example of several membership functions is given below, with a graphical interpretation in Figure 3.2. In this case there are three membership functions: cold, warm, and hot, describing temperature. Fuzzy sets such as these are seen as being a good way of applying natural language to database queries (Kollias & Voliotis 1991).

$$\mu_{\text{Cold}}(t) = \begin{cases} 1 & t < 8, \\ (12-t)/4 & 8 \leq t < 12 \\ 0 & 12 \leq t. \end{cases}$$

$$\mu_{\text{Warm}}(t) = \begin{cases} 0 & t < 8, \\ 1 - ((12-t)/4) & 8 \leq t < 12, \\ 1 & 12 \leq t < 16, \\ (20-t)/4 & 16 \leq t < 20, \\ 0 & 20 \leq t. \end{cases}$$

$$\mu_{\text{Hot}}(t) = \begin{cases} 0 & t < 16, \\ 1 - ((20 - t)/4) & 16 \leq t < 20, \\ 1 & 20 \leq t. \end{cases}$$

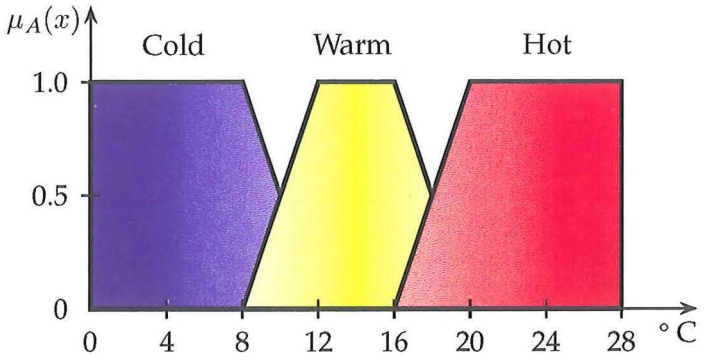


Figure 3.2: Fuzzy membership functions for temperature.

3.1.1 Non-normalised membership functions

The *height* of a fuzzy set is the maximum value that can be produced by the membership function. A normalised fuzzy set is one in which at least one element of the set reaches a membership level of 1. Not all fuzzy sets are normalised, although all example sets so far given in this thesis have conformed to that criterion. If the maximum membership value is less than one, the fuzzy set defined by the membership function is termed non-normalised. An example of a non-normalised fuzzy set is given in Figure 3.3.

3.1.2 Alpha cuts

An α-cut of a fuzzy set A is a crisp set A_α, that contains all the elements of the universal set X that have a membership grade in A greater than or equal to the specified value α. This definition can be written as

$$A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\}$$

An example of a 0.3 α-cut of Function (3.1) is given in Figure 3.4. α-cuts are used to transform fuzzy sets into crisp sets, and will be used in Chapter 6 to convert a surface

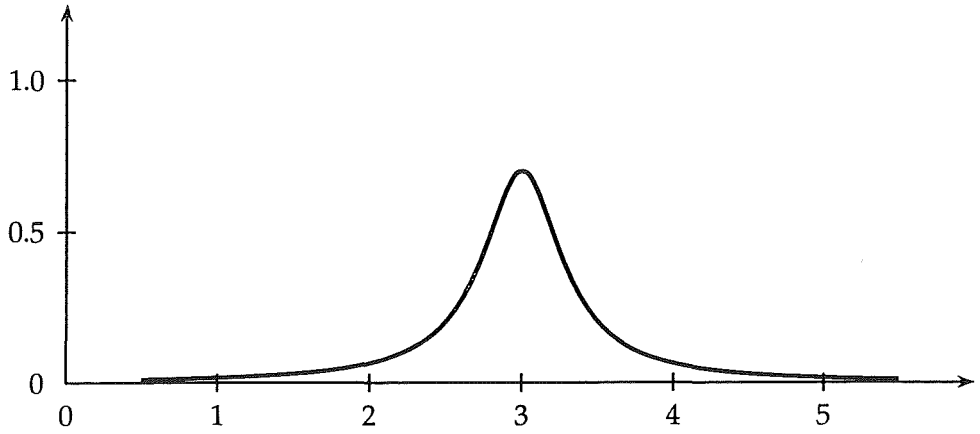


Figure 3.3: Non-normalised fuzzy membership function.

representing membership values into a collection of crisp polygons. In this way, features with a continuous variation and spatial distribution can be can be easily if not accurately approximated using a discrete data structure.

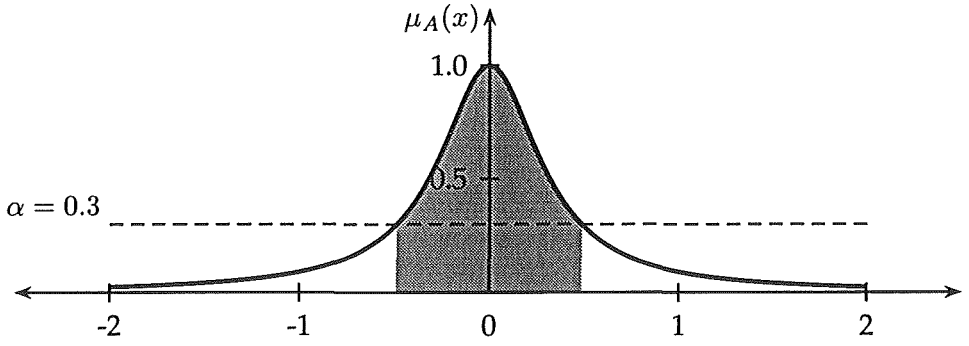


Figure 3.4: α-cut for $\alpha = 0.3$.

3.2 Classification of remote sensed images

Multi-spectral remote sensed satellite images are composed of several bands, one from each sensor on board the satellite. Each band corresponds to a different wavelength of energy. The number of bands present in an image can range from two up to the seven bands of a Landsat Thematic Mapper image. The bands are aligned so that a pixel in one band will correspond to the same pixel in each of the other bands. When the values of a pixel in each of these bands are combined, they define a spectral signature.

Many pixels can, and often do, have the same spectral signature. The shape of the signature depends on the wavelengths of energy that are reflected from the material on

the ground. It is possible to identify particular features based on their spectral signature. For example, water absorbs most of the near-infrared light striking it, while healthy vegetation reflects most near-infrared light. If signatures for two or more pixels are the same or similar, it is assumed that the ground material at the location represented by the pixels is the same. The comparison of spectral signatures forms the basis for land classification using remote sensed images.

There are many methods that can be used to classify pixels within an image. All these methods can be categorised into two broad types:

- Unsupervised, or
- Supervised.

Which of these approaches is the most appropriate to use for a classification depends on several factors, including: how much time is available to perform the classification; and how much information about the area to be classified is known prior to the classification process. An unsupervised classification is usually used if little prior information is available for an area, and time is limited. This type of classification requires very little user input as the classification algorithm usually calculates the spectral signatures for each class based on criteria such as maximising the distance between class signatures. Supervised classifications require that a user define a number of class signatures before the classification process begins. The class signatures are usually determined by locating known points on an image, and performing a field survey to determine what features are present at these locations. Alternatively, a radiometer may be taken into the field and used to identify the spectral signature of each feature of interest (Harris 1987).

Traditional classification techniques usually produce Boolean results. Each pixel in an image is classified into one and only one of several possible classes. In some cases a probability value indicating the accuracy of the classification is also calculated for each pixel. This accuracy measure is of little use if the next most likely class for the pixel is not known. Fuzzy classification differs from traditional methods in that not only is fuzzy logic used in the process of determining class signatures during an unsupervised classification, but also the classification results produced for each pixel are also in terms of fuzzy set membership rather than Boolean set membership. In this way, a pixel can have partial membership in all classes. A pixel's most likely class is the one in which it has highest membership.

3.2.1 Unsupervised classification

An unsupervised classification is executed by making two passes over an image. During the first pass, the spectral signature of every pixel is examined, and a spectral signature for each class is calculated. The number of classes that are present in an image is usually specified by the user, although there are methods that can be used to estimate the number of classes present. The method used for calculating class signatures is often based on a clustering algorithm, where each class is a cluster in the data. Clustering is discussed further in Chapter 4, with particular reference to the fuzzy c-means clustering algorithm. A popular approach in remote sensing is to identify clusters within a feature space so that the distance between clusters is maximised, and the distance between items in a cluster is minimised. Another approach to clustering is based on the selection of the most frequently occurring spectral signatures, and using these signatures as the estimates of cluster centres.

After the cluster centres are identified, a second pass is made over the image. The spectral signature of each pixel is compared to the signature of each cluster centre. The comparison can be done in one of several ways, some of which are: box classifiers, discriminant functions, and maximum likelihood classifiers (Harris 1987). The comparison is used to determine which cluster a pixel belongs to, and should therefore be classed as. Because clustering algorithms are based on statistical groupings, it is possible that some classes will not be correctly identified. The likelihood of an incorrect cluster occurring increases if there are several different features in an image that have similar spectral signatures.

3.2.2 Supervised classification

Supervised classification differs from unsupervised classification in that the user's prior knowledge, gained from previous studies or from field surveys, is used to determine spectral signatures for each class. The first step of an unsupervised classification can be avoided as the class signatures are defined by the user. The advantage of a supervised classifier is that the user has greater control over the definition of classes. In this way, class signatures can be customised to account for the situations that the statistically based cluster algorithms may not be able to detect. Once the spectral signatures for each class are defined, a pass is made over the image to classify each pixel. This pass is identical to the second pass made during an unsupervised classification.

3.2.3 Fuzzy classification

Fuzzy classifiers can be included in both the supervised and unsupervised family of classifiers. The fuzzy classifiers incorporate fuzzy logic at two levels. First, in the case of an unsupervised classification, membership levels are used to weight distances when calculating cluster centres. Second, membership values are calculated for each pixel to represent that pixel's level of membership in each class. The class in which a pixel has highest membership is the class to which that pixel is most likely to belong. The fuzzy c-means classification algorithm, a popular example of a fuzzy classifier, is described fully in Chapter 4.

3.3 Summary

This chapter has given a brief introduction to fuzzy logic, concentrating on those aspects that are relevant to fuzzy classification, and the representation of continuous information. By using membership levels rather than Boolean classification for pixels, significantly more information about the variation of features over a space can be stored and used for later processing.

Chapter 4

Measurement and Description of Continuous Spatial Data

This chapter describes a method by which information may be extracted from a remotely sensed image, and described in a manner suitable for further processing. In order to describe the spatial variation of features such as vegetation, the distribution of these features must be measured over the entire area of interest. A discussion is now given of issues concerning the choice of a data source for measuring spatially continuous variables, followed by a description of the classification algorithms used to derive the spatial distribution of features from the chosen data source.

The data source influences not only the type and amount of data available for processing, but also the types of processing that may be applied to the data. To detect natural variation, it is important to use data that have had little or no processing applied. For this reason, remote sensed satellite imagery has been chosen for this thesis as the most suitable data source. Section 4.1 describes the criteria by which this decision was made, and the technical specifications of the chosen imagery.

Information useful for measuring continuous variation is often lost when using traditional classifiers, but can be retained by applying fuzzy logic to the classification of pixels in a remote sensed image. Section 4.2 begins with an explanation of the difference between hard and fuzzy partitioning of data, followed by a description in Section 4.3, of the fuzzy c-means (FCM) classification method. A more efficient version of FCM, the approximate fuzzy c-means (AFCM) classifier, is presented in Section 4.4. Finally, Section 4.5 discusses some of the deficiencies of the FCM algorithms.

4.1 Data sources

Obtaining data of a continuous nature is difficult when working with information that has been heavily processed. Typically, a large amount of generalisation and approximation may have been applied to data collected in the field in order to input it into a GIS. On the other hand, remote sensed (RS) imagery contains a large amount of information over a continuous space, and apart from possible geometric correction, is free of processing error. Imagery from the French Satellite Probatoire d'Observation de la Terre (SPOT) was therefore chosen for use in this study. The technical specifications for SPOT follows.

Satellites:

- orbit at an altitude of 830km,
- remain synchronised with the sun,
- complete one revolution of the earth every 101 minutes,
- have a return period of 26 days,
- can perform stereo-scopic viewing of a scene.

Each image has:

- coverage area of 60 km $\times$ 60 to 80 km,
- a pixel resolution of 20m, and
- three spectral bands:
  - 0.50 - 0.59 μm (blue),
  - 0.61 - 0.68 μm (green),
  - 0.79 - 0.89 μm (near infrared).

The sensor in the SPOT multi-spectral (MS) scanner is a charge-coupled device (CCD) array that measures solar radiation reflected from earth. The area of coverage varies between images according to the viewing angle of the satellite. The viewing angle is controlled by a steerable mirror located in front of the CCD array.

SPOT images may be used for vegetation classification in areas that require a higher resolution than that provided by other satellites such as Landsat MS, which has a pixel resolution of approximately 80 square meters. Higher image resolution enables the detection of vegetation patterns at an increased level of detail.

The image used in the experiments to be described in this thesis was extracted from a SPOT MS image of Lyttelton Harbour in Banks Peninsula on the east coast of the South Island of New Zealand. The area covered by the 100 × 117 pixel image is that of Purau Bay (Figure 4.1), an area that for the purposes of this study includes three main classes: water, grass/scrub, and small stands of pine trees.

4.2 Clustering and classification

Cluster analysis and classification are two closely related concepts. Cluster analysis is concerned with the problem of partitioning a set X into c subsets or *clusters*, where c is some integer. It is expected that members of each cluster should exhibit greater similarity to each other than to members of another cluster, where similarity, in pixel classification terms, refers to the comparison of spectral signatures. Such signatures are explained in Section 4.3.

In traditional methods of clustering, subsets are mutually exclusive, and the union of all subsets produces the original set X. If the above conditions are met, then the subdivision of the set is referred to as a *hard* c-partition of the set X.

Cluster identification is based on the premise that data have *structure*. The structure of a data set is determined by the values of the characteristics of each data item within the set. Consider one characteristic of soil, soil temperature for example, where values can be used to partition a set of samples. Clusters are identified by finding cluster centres that fulfill two objectives:

- to minimise distances between data points within a cluster, and
- to maximise the distance between cluster centres.

Classification is achieved by assigning each data point to a cluster. Commonly used classification criteria are: nearest neighbour, maximum likelihood, and Mahalonobis distance (Bezdek 1981, Harris 1987).

Traditional clustering and classification methods are very coarse, and make broad generalisations about the data being classified. For example, each point in X can be a member of one class only, with no apparent similarity to other classes. Also, membership within a class is uniform, each member being as much a member as is any other member, regardless of how close each member is to the centre of the cluster.

Bezdek (1981) incorporated the idea of fuzzy sets into the clustering process. A fuzzy set is an extension of a classical set, the difference being that fuzzy set membership is

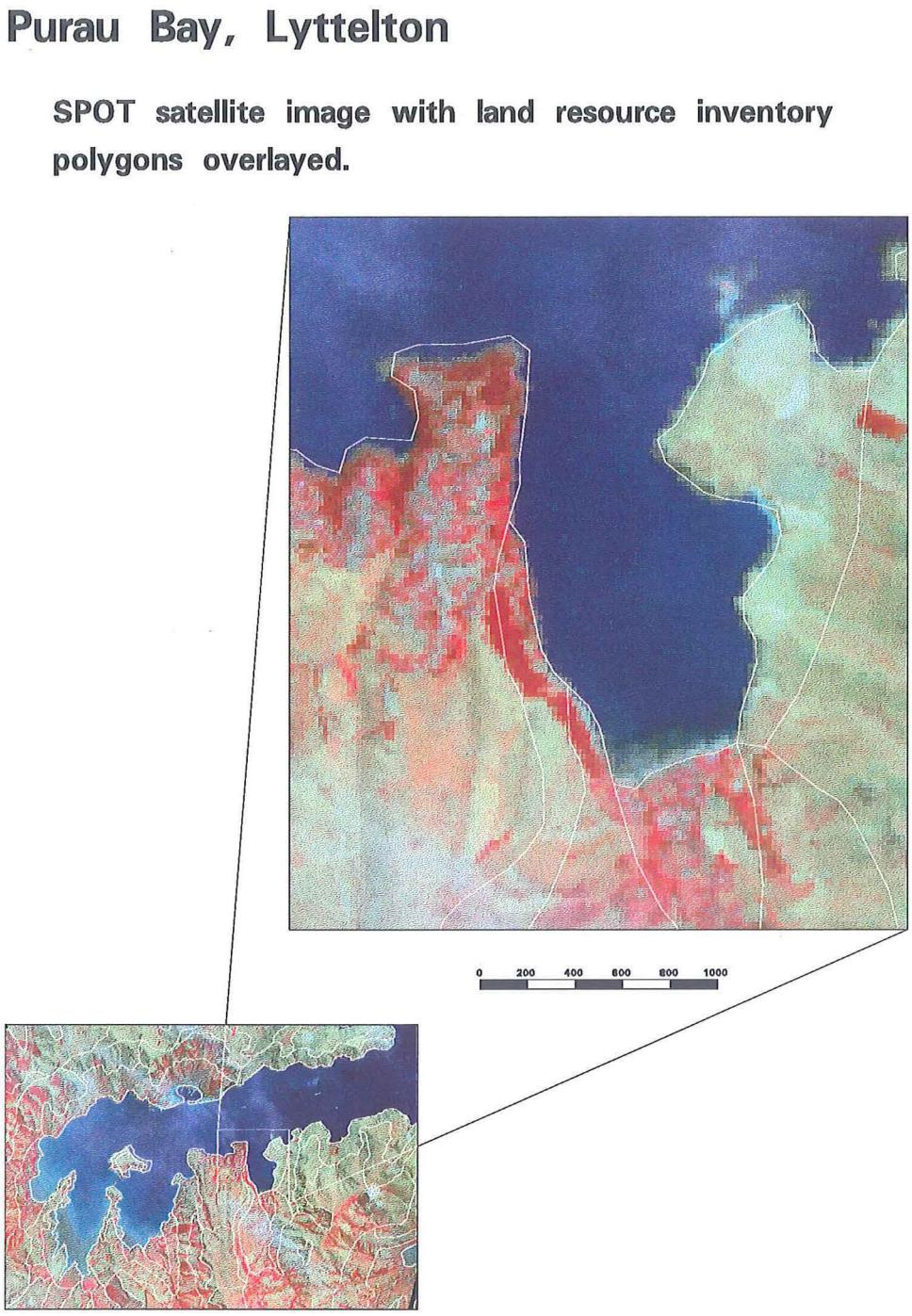


Figure 4.1: Study area.

measured within a range $[0, 1]$, rather than as a Boolean decision $\{0, 1\}$. Using fuzzy subsets to define clusters allows a member of the set X to have *partial membership* in each of the subsets, with membership being measured in the range $[0, 1]$. A membership value close to 1 indicates a high degree of membership, while a membership value close to 0 indicates a low degree of membership. The sum of the subset memberships for a given member of X must equal 1. The maximum membership value of all members of a given subset may not necessarily be 1, meaning that the membership function for that subset is non-normalised (Section 3.1.1).

The term fuzzy c-partitioning is used to describe partitioning algorithms that incorporate fuzzy set theory as part of the partitioning process. Fuzzy c-partitioning does not involve the generalisations associated with hard or Boolean partitioning. Information about each class is not restricted to the pixels of the image where the classification algorithm identifies the pixel as belonging to the class, the degree of membership in each class is calculated for each pixel across the entire image.

4.3 The fuzzy c-means classifier

The fuzzy c-means (FCM) algorithm is a popular and well studied example of a fuzzy classifier (Cannon, Jitendra & Bezdek 1987, Wang 1989, Wang 1990, Pathirana 1992). In this thesis the FCM algorithm is used to classify pixels within a SPOT MS remote sensed image.

A SPOT MS image has three component images, one for each spectral band sensed by the satellite. The location and area of land covered by a given pixel is the same for all three bands. Hence, pixels have three values corresponding to the spectral reflectance measured by each of the three sensors on board the satellite. Together, pixel values form a spectral signature describing the feature located at the position of the pixel. Figure 4.2 shows a spectral signature curve representing the reflectance at various wave-lengths for healthy vegetation. The three bands measured by the sensors on board the SPOT satellite are overlaid in grey.

The FCM algorithm uses each pixel's spectral signature to find clusters of like-valued pixels in the image. Clusters are located by plotting signatures in a *feature space*, which is defined by the bands in the image. The number of bands present in the image determines the dimension of the feature space, while the maximum and minimum reflectance of each band determines the extent of the feature space.

For example, the feature space for a SPOT MS image is three-dimensional because

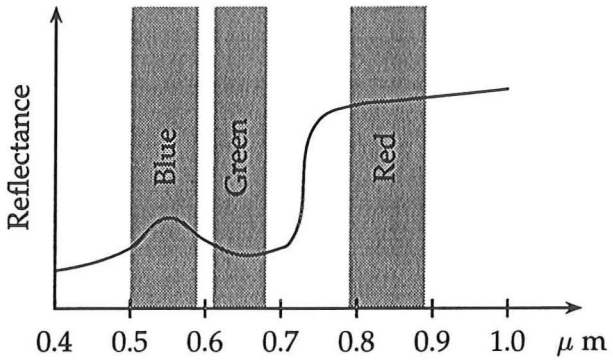


Figure 4.2: Spectral signature for healthy vegetation (Harris 1987).

there are three separate bands, with an extent of $[0, 255]$ on each axis. When each pixel is located in that feature space, the FCM algorithm finds cluster centres within the feature space such that the two objectives described in Section 4.2 are fulfilled.

Once cluster centres are found, pixels are classified to determine the extent of their membership in each cluster. The FCM algorithm calculates the distance between a pixel and a cluster centre in the feature space. The distance is then used to determine the level of membership a pixel has in the cluster: the greater the distance to the cluster centre, the lower will be the value for membership in that cluster. If there are three clusters in the feature space, a pixel will have three membership values, one for each cluster.

The FCM algorithm requires a user to supply approximate spectral signatures for the cluster centres. These approximate centres are a starting point for the algorithm, and they change as the algorithm calculates an optimal partitioning of the image. The optimal clusters identified by the FCM algorithm will represent similar features to those specified by the initial signatures. The requirement to provide initial cluster centre signatures is not to be mistaken for the FCM algorithm being a supervised classifier described in Section 3.2.2. Random numbers may be used as initial signatures for cluster centres, and will lead to the same results as if optimal centres had been used.

The computation time and memory space required by FCM can make the process impractical for real time use with large images and many clusters. Section 4.4 describes a modified version of FCM (Cannon et al. 1987) that considerably reduces computation time and memory requirements for classification.

The amount of data produced by the FCM algorithm can become a problem, as significantly more output is generated than from a hard classifier. A hard classifier produces a single value for each pixel in an image, a reference to the class to which the pixel belongs. The FCM algorithm produces a membership value for each pixel in each

cluster. As a consequence, if there are c clusters, FCM produces c times as much data as a hard partitioning algorithm. The solution to this problem used in this thesis is described in Chapters 5 and 6.

4.3.1 Overview of the FCM process

The FCM algorithm uses an objective function that defines the criteria against which a possible partitioning is measured. The objective function used by the FCM algorithm is based on the least squares method for linear regression. This function is a formal interpretation of the two clustering objectives described in Section 4.2. Minimising the objective function results in an optimal partitioning of the signatures. The algorithm repeatedly clusters and classifies pixels in an image, moving the cluster centres until the function is minimised. The differences between the FCM objective function and the least squares method are discussed in Section 4.3.2.

The FCM algorithm begins by loading a sub-sample of a SPOT MS image from a LAN file (Erd 1991), the native format for Landsat data and the Erdas image processing software package. Initial signatures for the approximate cluster centres are defined using the tools provided by Erdas, and read from an Erdas signature binary description (SBD) file (Erd 1991). The next step performed by the algorithm is to randomise the level of membership for each pixel in each cluster. This is done by calculating a series of uniformly distributed random numbers, and assigning them to the membership values for each pixel in each cluster.

Having completed the initialisation stages, the algorithm begins an iterative process of minimising the objective function. Two equations are used to obtain a local minimum for the objective function. The first is used to calculate cluster centres; the second is used to calculate the membership values, in effect classifying each pixel. A local minimisation of the objective function is one which, given one set of membership values, will produce a minimal result from the objective function. Achieving a local minimisation is not sufficient and the process must be repeated until a steady state is reached where membership values remain constant from one iteration to the next.

The two equations that minimise the objective function form the basis for optimising the cluster centres and are described in Section 4.3.2. A third step calculates the maximum change in membership level for each iteration. When the maximum change drops below a predefined level the membership values are assumed to have reached a steady state, the objective function is minimised, and the partitioning of the feature space has been optimised.

The levels of membership for each pixel are output as grey-scale raster images. One image is produced for each cluster, and a pixel in the image represents the same area covered by the corresponding pixel on the original SPOT image. The value of a pixel represents the level of membership of that area to the cluster associated with the image to which the pixel belongs.

The flow diagram in Figure 4.3 illustrates the overall FCM process.

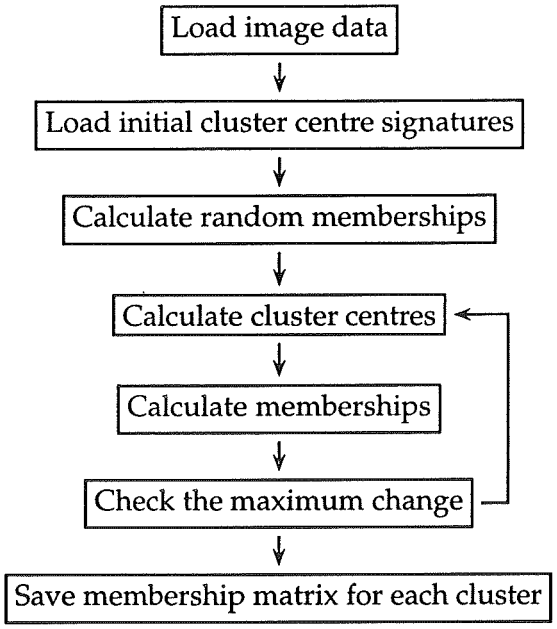


Figure 4.3: Flow diagram of FCM algorithm.

Reversing the order in which cluster centres and membership values are calculated removes the need to randomise the membership values before performing the first iteration. Reversing the order can be achieved by calculating approximate values for the membership values rather than using random values. Experimental results in Table 4.1 show that, for certain data, calculating membership values first results in fewer iterations being needed to achieve an optimal partition. Two factors were found to influence the number of iterations required to achieve an optimal partition:

- the existence of clusters in the data,
- the choice of initial cluster centres.

As discussed in Section 4.2, clustering assumes the existence of structure in the data being clustered. Data that are uniformly distributed with no distinct clusters require a

greater number of iterations to produce an optimal partition than is required for data that contain obvious clusters. When partitioning data with indistinct clusters and calculated initial membership values, the choice of initial centres can significantly increase or decrease the number of iterations required to reach an optimal partition.

Table 4.1 shows the results obtained from clustering a set of random numbers using the FCM algorithm. The first major column headed "Structured" refers to the distribution of the random numbers to be clustered. The structured data were generated by applying a function to uniformly distributed random numbers. The function modified the distribution of the random numbers, causing two distinct clusters to appear in the data. The non-structured data were generated from the same uniformly distributed random numbers, without applying the distribution modifying function, thus ensuring that there were no clusters. Within each of the major columns there are two sub-columns labelled "Random" and "Calculated". These labels refer to the method used to compute the initial membership values for each pixel.

The FCM algorithm was repeatedly executed with the same data set, but different initial cluster centres, until all possible centres had been used. The aim was to observe what effect varying the initial cluster centres would have on the number of iterations required to reach an optimal partition of the data set. For each set of cluster centres the number of iterations required was recorded. The maximum, minimum, and average number of iterations required to cluster the data are presented in the table below.

	Structured		Not Structured	
Iterations	Random	Calculated	Random	Calculated
Minimum	7	2	9	4
Maximum	7	6	9	10
Average	7	5.94	9	8.63

Table 4.1: Comparison of the orderings for the steps in FCM.

It can be seen from these results that, when initial membership values are generated randomly, the number of iterations is constant, independent of the initial cluster centres provided by the user. In the case where initial membership values are calculated before the clustering process begins, the initial cluster centres can influence the number of iterations required to reach an optimal partition. When the data being clustered have no distinct clusters, and the initial cluster centres provided by the user are significantly different from the optimal centres, calculating initial membership values based on the erroneous cluster centres may result in an even greater number of iterations than if the membership

values were random.

4.3.2 Optimal partitioning with FCM

As discussed in the previous section, the FCM algorithm uses an objective function based on the weighted least squares function. Given that there are N pixels to be classified into c clusters, multiple passes are made over the data, each time refining the cluster centres to minimise the objective function:

$$\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik})^m(y_k - v_i)^2 \tag{4.1}$$

where u_{ik} is the level of membership for the kth pixel, y_k, in the ith cluster, v_i. The variable m is an *exponent weight*, which can be used to "tune out" noisy pixels from the objective function. If a pixel has a low level of membership for all clusters its contribution to the objective function can be reduced by increasing the value of m. It should be remembered that y_k and v_i are vectors defined by their spectral signatures in the feature space of the image, and so $y_k - v_i$ is also a vector.

A local minimum of the FCM objective function can be computed for given $u_{11} \ldots u_{Nc}$ and $v_1 \ldots v_c$ with the following two equations:

$$v_i = \sum_{k=1}^{N}(u_{ik})^m y_k / \sum_{k=1}^{N}(u_{ik})^m; \quad 1 \leq i \leq c \tag{4.2}$$

$$u_{ik} = \left(\sum_{j=1}^{c}\left(\frac{y_k - v_i}{y_k - v_j}\right)^{2/(m-1)}\right)^{-1}; \quad 1 \leq k \leq N; \quad i \leq i \leq c. \tag{4.3}$$

These equations result in new values for the cluster centres v_i, and the membership values u_{ik}. By repeatedly applying (4.2) and (4.3), a state will be reached where the membership values for each pixel remain constant between iterations. When the algorithm reaches this steady state, the resulting cluster centres and membership values provide a minimum for the objective function (Windham 1982).

Calculating cluster centres is similar to calculating the centre of gravity for a set of masses, except that membership values are used rather than masses, and distance is measured in terms of the difference between spectral signatures of the pixels. The difference between spectral signatures is measured in terms of vectors within the feature space.

Membership values are inversely proportional to the distance from a pixel to a cluster centre relative to the distance to all other clusters as plotted in the feature space. Hence, equation (4.3) is similar to that used for calculating weights for linear interpolation.

4.4 The approximate fuzzy c-means classifier

A problem associated with the FCM algorithm is the large amount of computation required to find the cluster centres and to classify each pixel. For each pixel, the distance to each cluster centre must be calculated, followed by several exponentiations to determine cluster membership. The *approximate FCM* (AFCM) algorithm (Cannon et al. 1987) was designed to avoid the problem of calculating many computationally intensive formulae. The AFCM algorithm reduces the number of calculations performed per iteration by creating several look-up tables to replace the most costly calculations.

Three calculations were identified by Cannon et al. (1987) as having the greatest effect on overall classification time:

- $y_k - v_i$, the distance from a pixel y_k to a cluster centre v_i, as measured in the multi dimensional feature space.
- u_{ik}, the membership value for pixel y_k in cluster i.
- v_i, the centre of cluster i within the feature space.

A number of assumptions were made by Cannon et al. (1987), namely that (i) the source data (reflectance values) would take discrete values from the set $0 \ldots 255$, (ii) that the precision of cluster centres $v$were limited to one decimal place multiplied by 10 to ensure that the tables were limited to a tractable size; and (iii) membership values u_{ik}, which are normally in the range $[0.0, 1.0]$, were rounded to three decimal places and multiplied by 1000 to produce approximations in the range $[0, 1000]$. These assumptions were reflected as constants in the equations used to calculate the tables.

For the sake of simplifying the description of the tables, these constants have been removed in the following discussion, and the tables as defined below differ slightly from those described by Cannon et al. (1987). Six tables are used to approximate the equations required for the three major calculations. Of these tables, one must be recalculated for each iteration of the clustering algorithm, and the others are calculated once, before the iteration process begins. The AFCM algorithm also differs from FCM in that cluster centres are updated after membership values are computed. The following three sections describe the six tables in greater detail.

4.4.1 Approximate $y_k - v_i$

Two tables are used to approximate $y_k - v_i$, the distance from a pixel to the cluster centres. This distance can be in one or more dimensions. Because the cluster centres move with

each iteration of the loop, the first of these tables must be calculated at the beginning of each iteration. If v_{il} is the value in the lth dimension (band) of the ith cluster centre, the squared difference between y and v_i in band l is stored in the table:

$$\text{Table\_A}[i, l, y] = (y - v_{il})^2; \quad 1 \leq i \leq c; \quad 1 \leq l \leq p; \quad 0 \leq y \leq 255.$$

The second table replaces the square root calculation in the distance component of Equation (4.3). Values obtained from Table A for each dimension are summed and normalised by p. Normalising by p reduces the necessary size for Table B. Assuming most distances will be less than or equal to 10000, square roots for these values are stored in Table B, while square roots for distances greater than 10000 are calculated by calling a library function:

$$\text{Table\_B}[y] = \sqrt{y}; \; 0 \leq y \leq 10000.$$

By combining Table A and Table B in Equation (4.4) the distance between a pixel and a cluster centre in the feature space can be calculated from $p + 1$ table look-ups and one division operation:

$$y_k - v_i = d_{ik} = \text{Table\_B}\left[\frac{\sum_{l=1}^{p} \text{Table\_A}[i, l, y_{kl}]}{p}\right]. \tag{4.4}$$

4.4.2 Approximate u_{ik}

In order to calculate the membership value for one pixel in one cluster, a division is performed for every cluster in the data set. The division operation can be replaced by the difference of the logs of the two numbers being divided, and raising 10 to the power of the result. Table C provides the logarithm values to be subtracted, and Table D provides values for 10 raised to the power of suitable numbers such that the range of possible results from the subtraction is covered. It is assumed that m will always be greater than 1.01.

$$\text{Table\_C}[y] = (2/(m-1)\log y); \; 0 \leq y \leq 255$$

$$\text{Table\_D}[y] = 10^y; \; -200 \leq y \leq 482.$$

Membership values, u_{ik}, can now be calculated with one division operation and $2c+1$ table look-ups:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \text{Table\_D}[\text{Table\_C}[d_{ik}] - \text{Table\_C}[d_{jk}]]}.$$

4.4.3 Approximate v_i

A literal interpretation of the FCM classification method requires N exponentiation and multiplication operations to calculate one coordinate for a cluster centre. To calculate the location of all cluster centres in the feature space would require $N.p.c$ exponentiation and multiplication operations. Table E provides logarithm values which can be used to replace $(u_{ik})^m$ from Equation (4.2). The value x_{kl} is replaced by a corresponding value obtained from Table F:

$$\text{Table\_E}[y] = m \log(y);\ 0 \leq y \leq 1$$

$$\text{Table\_F}[y] = \log(y);\ 1 \leq y \leq 255.$$

Together the values obtained from Tables E and F are used to calculate an index for Table D. These three table look-ups and one addition operation replace the exponentiation operation that formed the numerator of Equation (4.2). The denominator of Equation (4.2) is similarly replaced by two table look-ups. In contrast to the literal interpretation of Equation (4.2), the AFCM interpretation replaces two exponentiation operations with one addition operation and four table look-ups:

$$v_{il} = \frac{\sum_{k=1}^{N} \text{Table\_D}[\text{Table\_E}[u_{ik}] + \text{Table\_F}[y_{kl}]]}{\sum_{k=1}^{N} \text{Table\_D}[\text{Table\_E}[u_{ik}]]}.$$

Cannon et al. (1987) obtained results that showed AFCM is approximately six times faster than FCM. The data sets used for the comparison were a collection of 256 × 256 pixel images, composed of nine bands. The FCM algorithm was used to identify 10 clusters for which each pixel's level of membership was calculated.

Figure 4.4 shows several results obtained from the classification of the Purau image using an implementation of the AFCM algorithm. The images show the variation that resulted from changing the value of m. The image was classified into three classes, resulting in three membership rasters. The colour of each pixel is calculated by mapping the membership value for each class to a particular colour component. The pine class is mapped to red, the grass class is mapped to green, and the water class maps to blue. When m is equal to 1.1, the classification is very "crisp", with each pixel being one of the three primary colours: red, green, or blue. Increasing m has the effect of reducing the cluster sizes in the feature space. This reduces the level of membership in each cluster for each pixel, except those pixels that are very close to a cluster centre, causing more detail of the variation to be revealed.

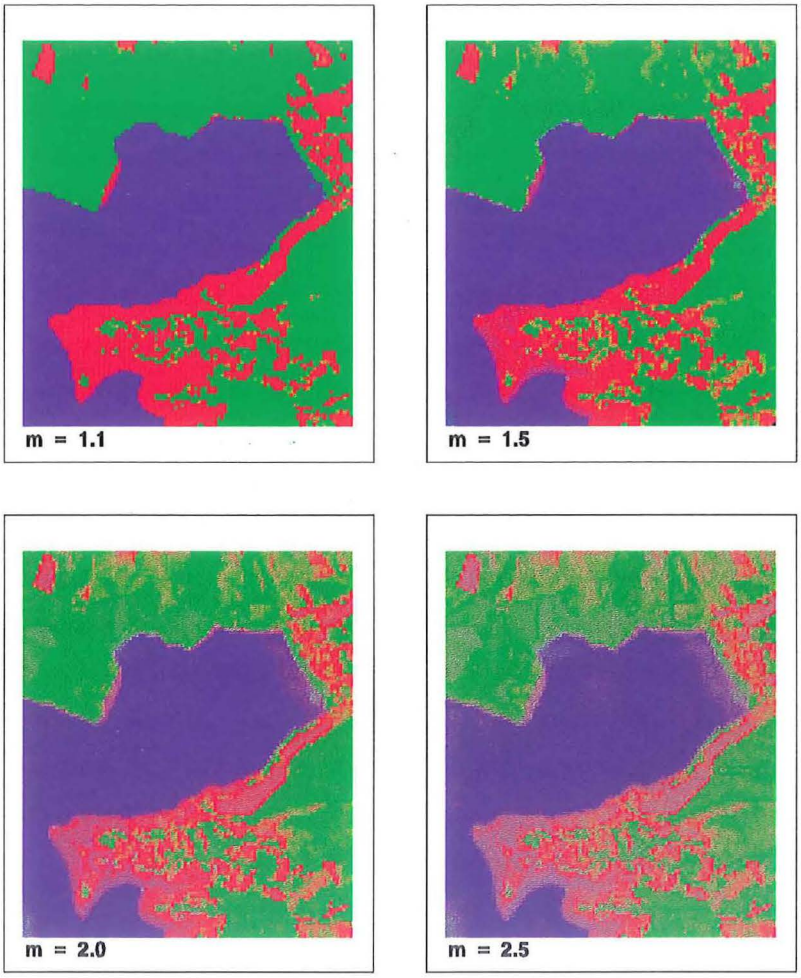


Figure 4.4: Example of varying the exponent.

4.5 Summary

- The FCM and AFCM algorithms are designed to locate cluster centres, and to calculate membership values for each pixel based on the distance separating the pixel and the cluster centre in the feature space. Because membership values are based on distances within the feature space, clusters are assumed to be circular in shape when the feature space is two dimensional, and spherical when the feature space is three dimensional. FCM and AFCM therefore produce inaccurate membership values when clusters form shapes other than circles or spheres. Attempts have been made by Bezdek (1981) to use fuzzy sets to measure within-cluster variability, thus providing a way of modelling cluster shape.

- The AFCM algorithm can be used to to perform a supervised classification, as described in Section 3.2.2, where optimal cluster centres are not calculated, and pixels are classified relative to the initial cluster centres provided by the user. If only one pass is performed over the image, the cluster centres that are provided by the user are not modified. Hence the resulting membership values are based entirely on the cluster centres. Note that this will only produce reliable results if the algorithm calculates membership values before updating cluster centres.

- The value of m determines the *contrast* of the classification. By changing the value of m, it is possible to move from a hard classification ($m = 1$) to increasingly fuzzy classifications. As m increases, pixels with similar levels of membership in all clusters have less effect on the calculation of membership values and cluster centres. Pixels that have higher levels of membership in a particular cluster have greater influence as m is increased. This has the effect of decreasing the extent of each cluster, leading to noisy results as pixels no longer clearly belong to one or other clusters. Commonly accepted values for m range from 1.5 to 2.0 (Bezdek 1981, Bezdek et al. 1984, Cannon et al. 1987, Pathirana 1992). As m increases above 3 the resulting classification becomes too noisy to be of use. A value for m must be chosen that will allow enough variation in membership values to identify the change that occurs at boundary regions, without allowing so much variation that removing redundant pixels may be difficult.

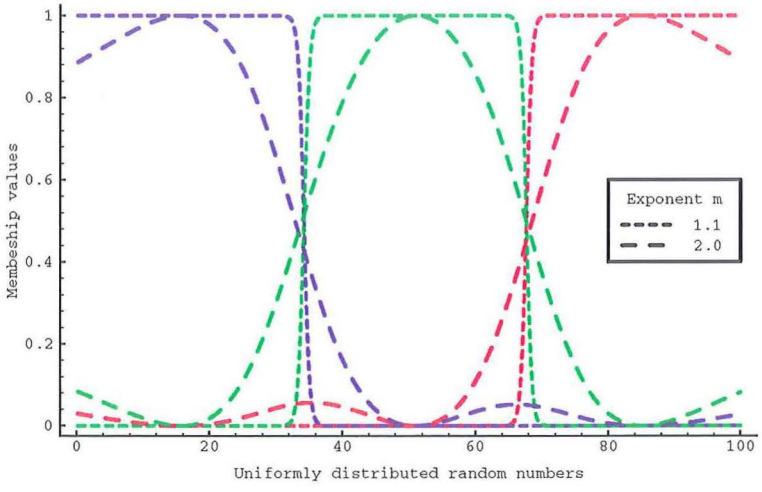


Figure 4.5: Effects of varying m.

Figure 4.5 shows the membership values for uniformly distributed random numbers

between 1 and 100 after they have been clustered into three classes using FCM. Two clusterings were performed in this research, the first with $m = 1.1$, and the second with $m = 2.0$. The reduction in the extents can clearly be seen in the way that membership values drop away faster as the value for x moves away from the cluster centres.

- Section 3.1 discussed how membership functions can be defined and plotted for fuzzy sets. The membership functions for each of the three clusters identified by the FCM algorithm in the Purau image can also be plotted. A pixel's level of membership in each cluster is determined by the three values that make up the pixel's spectral signature. By plotting each pixel according to its membership in each of the clusters, the resulting graph in Figure 4.6 is comparable to those shown in Section 3.1. The colour of each pixel is determined by mapping the pixel's level of membership in the pine cluster to the red component, mapping membership in the grass cluster to green, and mapping membership in the water class to blue.

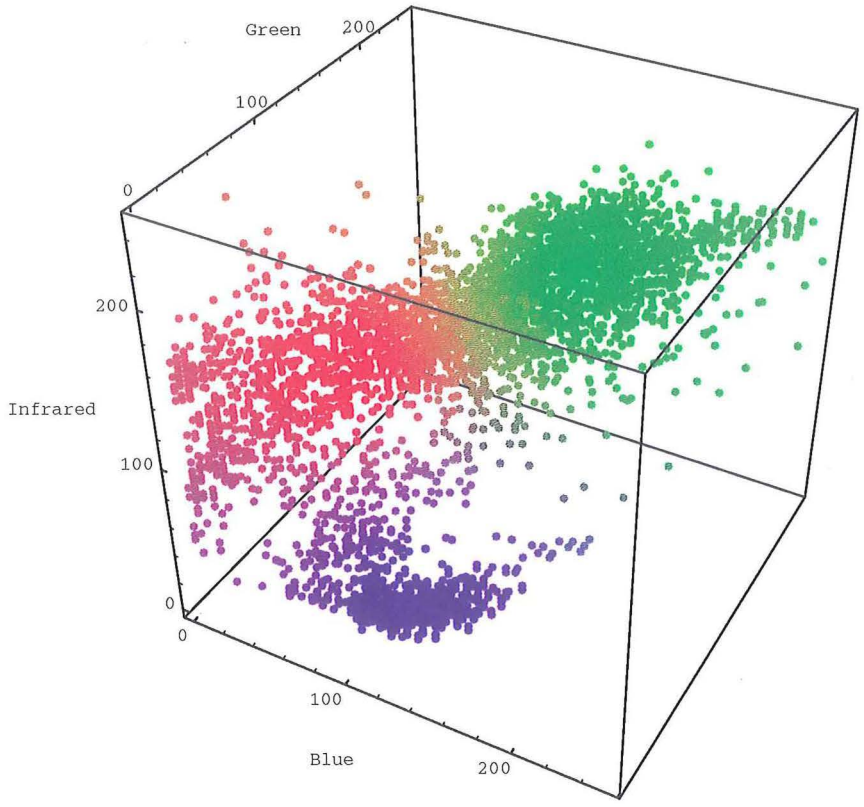


Figure 4.6: Example AFCM results ($m = 2.0$).

The change from one class to another class can be seen by the gradual change of colour from one cluster to another. The colour of a pixel is indirectly related to its spectral signature by way of the classification algorithm.

- Vast amounts of data are produced by the FCM family of classifiers (Leung, Goodchild & Lin 1992). Chapter 5 describes several approaches to reducing the amount of data to a manageable size by identifying critical information. Chapter 6 will discuss alternative methods of representation for the critical data.

Chapter 5

Reducing Data Volume Without Losing Information

This chapter describes four methods by which data obtained using the FCM algorithm can be *filtered* to remove information that is of little or no use. The FCM algorithm produces membership values for each pixel in a satellite image. Each pixel has a level of membership in every class that the user defines. One *membership raster* is therefore produced by the FCM algorithm for each class. Where there are large areas in the original image that are of the same class, the FCM algorithm will produce membership values that are the same or similar for each pixel in the membership raster. The pixels at the boundaries of regions will exhibit greater variation in their membership values.

The previous chapter finished by describing several problems with the FCM algorithm, including that of the large volume of storage space required. This chapter continues that theme by describing various ways of reducing the data produced by the FCM algorithm while retaining the important information. Chapter 6 describes a suitable data structure for storing the *critical* data points identified by the methods described in this chapter.

Each raster produced by the FCM algorithm has similar characteristics to the raster of a digital elevation model (DEM). In both DEM and membership rasters, each pixel has assigned to it a single value: height in the case of a DEM raster; a membership value in the case of a raster produced by the FCM algorithm. Maintaining this layer of values for every class used in the FCM classification requires too much storage space. In order to help overcome this problem each FCM-generated raster can be represented by a triangular irregular network (TIN).

A TIN allows irregularly spaced points. When used for a terrain model, a TIN has many points in areas of rough terrain and few points in areas of smooth terrain.

For a membership raster, a TIN will require few points where there are large areas of the same class, and many points at the boundaries of regions. Because surfaces can usually be represented with a TIN using many fewer points than a raster-based DEM, the TIN provides a more efficient form for storage. Similar savings should be possible for membership rasters. Existing methods that have been developed for use with TINs can also be exploited for manipulating membership values. Methods used to reduce a raster to a TIN are now described.

5.1 What is important information ?

The FCM classifier produces n membership rasters, where n is the number of classes, resulting in n times as much data as those classifiers that place a pixel into one of n possible classes. Some form of data reduction is required to remove redundant information, and decrease computer memory requirements. First, some method must be used to identify the points to be included in the TIN. The question of which pixels in a membership raster should be considered worthy of inclusion depends on several factors:

- the source data structure used to represent membership values,
- the data structure used for the final representation,
- the type of processing to be performed on the final representation.

The source data, in this case a membership raster, provide a grid of membership values. This format influences the way in which important pixels can be differentiated from those which are not important. For example, techniques used for image processing can be adapted and used on membership rasters because the rasters are essentially grey scale images. The data structure into which information is to be placed will affect what is considered important, and how to decide if a specific datum is important. Finally, if the way in which important information is identified is not considered during later processing the results can be unpredictable.

Having decided to represent each membership raster using a TIN, the term "important information" can be defined more precisely. A TIN is composed of many points, each being a vertex of one or more triangles. Few points are required to represent areas of gradual change. As the surface described by the membership values becomes more variable, the number of points required to represent accurately the variation increases. Constructing a TIN requires location of *critical points* or pixels in the membership raster

such that, when used to form vertices in a TIN, there will be a good approximation to the original membership surface.

5.2 Information theory

Information theory has been used in the field of text compression to calculate the information content of data. In this chapter the possibility of applying information theory to spatial data is considered, not for the purpose of compression, but also for identifying critical points from a membership raster. These critical points are then used as the vertices to form a TIN, while other points are discarded.

For text compression, information theory has been used to calculate how many bits should be used to represent each character. This calculation is based, for each character, on the probability of that character occurring in a message. Fewer bits should be used to represent characters that are likely to occur, while characters that occur only rarely should be represented using a greater number of bits. The less likely a character is to occur, the greater the amount of information that the character conveys.

This theory can be applied to the problem of locating critical points in a membership raster. For example, if a particular membership value is present only once in a large membership raster, then the pixel could be said to contain a high level of information and should therefore be used as a vertex in the triangulation of the surface. Alternatively, the value of the pixel could be considered to be noise, due to a poor classification, in which case labelling it as a critical point would be incorrect. Poor classifications are usually the result of setting the value of the weighting exponent m, greater than 3.0. Based on a survey of relevant literature, there is no method currently available to determine if a pixel is noise or real information. In some cases a membership value that occurs infrequently may represent a discrete feature such as a building. In other cases infrequently occurring membership values could be some form of noise. Section 4.5 includes a discussion on choosing a value for m.

5.2.1 Entropy

Given the set of all possible membership values between 0 and 255, with associated probabilities $p_0, p_1, \ldots, p_{255}$, the *entropy* E of this set measures how much choice is involved in the selection of one membership value from the set (Bell, Cleary & Witten 1990). If probabilities are skewed so that one membership value is highly likely to occur and all other membership values are unlikely to occur, then there is very little choice available.

If membership values occur with equal probability then the amount of choice is at a maximum.

The equation

$$E = -\sum_{i=1}^{N} p_i \log_2 p_i \tag{5.1}$$

where N is the number of possible membership values, and p_i is the probability of membership value i occurring in the membership raster, can be used to calculate the average entropy for a pixel. The average entropy is a measure of how much choice is available for the pixel. If one membership value has a high probability of occurring, and all other membership values have low probabilities, then the amount of choice is low. If all membership values have equal probability of occurring, then the level of choice is at its greatest. Probabilities for each membership value can be estimated by maintaining a frequency count of how many times each membership value occurs over the whole membership raster. Because the probabilities are calculated over the entire raster, the entropy will be the same for each pixel.

Of more interest than the average entropy for each pixel is the entropy of each membership value. The entropy, or amount of information contained in a single membership value i, can be calculated using

$$E_i = -\log_2 p_i \tag{5.2}$$

where p_i is the probability of membership value i occurring. Given a membership value for a pixel, Equation (5.2) can be used to calculate how much information that pixel contains, given that its membership value is i. By the definition of entropy, the greater the probability of an event, the less information it contains, while rare events contain greater information (Bell et al. 1990).

To calculate the entropy for each pixel in a membership raster, two passes are made over the image. During the first pass, a frequency table is built, with an entry for each possible membership value. In the case of the output from the FCM algorithm, the table contains 256 entries, covering the range [0,...,255] of possible membership values. Each time a membership value is encountered during the first pass, its counter is incremented. A second pass over the image is then made, calculating the entropy of each pixel using Equation (5.2) and the probabilities calculated from the frequency table. An output raster is then produced in which each pixel contains the level of entropy of the corresponding pixel in the input raster.

Figure 5.1 shows the output generated from the entropy calculation method. The first raster in the figure is one of the membership rasters generated using the AFCM

algorithm. Light pixels represent areas of high membership to the "pine" class, while dark pixels represent low levels of membership in that class. The second raster represents the probability of each pixel's membership value occurring in the membership raster. Pixels with membership values that occur often appear light, while pixels with less common membership values appear dark. The final raster depicts the information content of each pixel. Light pixels have high entropy, dark pixels have low entropy.

(a) Original (b) Probability (c) Entropy

Figure 5.1: Three stages of the entropy calculation.

5.2.2 Context modelling

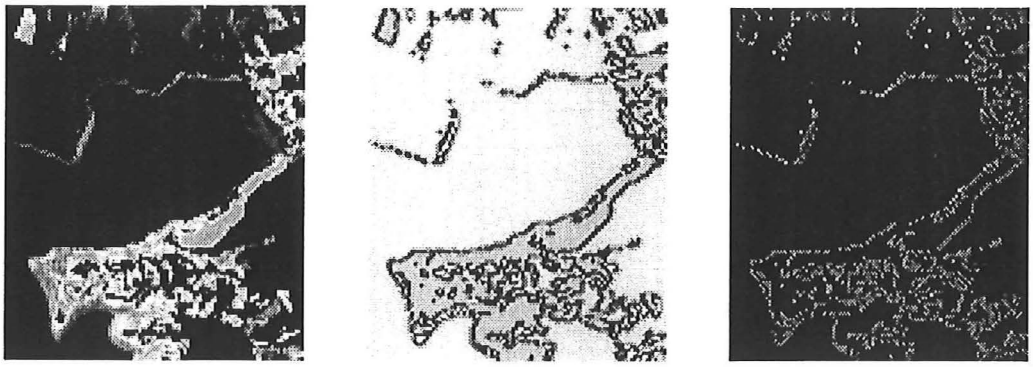
In the previous section, it was shown how the probabilities for each membership value could be used to calculate the information content of each pixel, with the probabilities taken from a frequency count over the entire membership raster. However, the probability of a pixel having a particular membership value is influenced by the membership values of surrounding pixels (Bell et al. 1990). For example, if a pixel in the membership matrix is surrounded by eight pixels with membership values of 0.75, it is highly probable that the membership value for the central pixel will be 0.75.

In the field of text compression, Bell et al. (1990) define an *order-0 fixed-context model* as a model where zero preceding characters are used to predict the next character. The example used in the previous paragraph could be described as an order-8 two-dimensional fixed-context model, because the eight surrounding pixels are used to predict the central pixel's membership value. Using a context model causes the distribution of probabilities for each membership value to be skewed.

There have been many approaches to image compression using context models. For example, Helman & Langdon (1988) have shown how a simple order-1 context model can result in better predictions for pixel values in a grey scale image. Also, Todd, Langdon & Rissanen (1985) and Moffat (1991) applied several different high order context models to produce significantly better predictions for similar grey scale images.

A problem associated with using context models with grey scale images is the choice of context size. For example, using a context of the four adjacent pixels, each having a value in the range [0,...,255], results in 256^4 possible sets of values for the context, though many of these combinations may never occur in the image. Helman & Langdon (1988) avoid the problem of building very large context models by using *buckets*, with the possible range of membership values split into discrete groups. If the range is subdivided into five buckets, there will be only 5^4 possible values for the context.

Figure 5.2 is similar to Figure 5.1, except that the probability and entropy for each pixel is calculated relative to the pixel's surrounding context. By using a context to calculate probabilities for each pixel, changes from areas of high membership to areas of low membership can be identified.



(a) Original (b) Probability (c) Entropy

Figure 5.2: Three stages of the context-based entropy calculation.

5.3 Surface trends

Methods derived from information theory identify information based on probabilities. This is appropriate for encoding information for compression, but was found in the context of this thesis to be less useful for encoding information in an alternative data

structure such as a TIN. The reason is that pixels with high levels of information in terms of entropy, are not necessarily those that are needed for building a TIN. For example, a membership raster may contain an area of gradual change over three pixels, represented by the following membership values: 25, 130, and 235. The membership value of 130 may only occur once in the entire raster, and as such would in theory contain a high level importance. Assuming membership values 25 and 235 are also critical points though, the middle pixel is redundant if a TIN is used to represent the gradual change.

Context modelling can add a spatial component to information theory, as the level of information for a pixel is influenced by its neighbouring pixels. The idea of a context is similar to the use of a window in the moving average interpolation process (Burrough 1986). The context size affects the scale at which critical points are detected. Using a small context should identify critical points that will produce good approximations of small patterns. Contexts that include pixels not adjacent to the central pixel, lose the detail of variation in the intermediate pixels. Reducing the context size and the range of each value forming the context will result in fewer context parameters. However, large generalisations occur when values in the range of 0 to 255 are transformed to the range 0 to 4.

Context models will adapt to the raster being scanned. If a raster contains a high proportion of pixels with membership values that change sharply from one pixel to the next, the entropy of a pixel with a membership value similar to the pixel's context values is very high. Information theory and context modelling cannot, however, be guaranteed to identify those pixels that will produce a close approximation when used to build a TIN.

5.3.1 In-betweening

Some of the ideas gained from context modelling, in a way that is useful for identifying those points to incorporate into the TIN. The concept of neighbouring pixels influencing the membership value of each pixel can be used in a slightly different way, with specific consideration of the TIN data structure.

If three pixels are taken in sequence, it would be expected, with a high level of probability, that the membership value for the central pixel would fall between the membership values for the two neighbouring pixels. If the value of the pixel is not between the two adjacent values, then the pixel represents a local maximum or minimum. Pixels that are local maximums or minimums represent a significant change in surface slope, and as such are critical points in the membership raster. Other pixels, whose membership values fall between those of their neighbours, can be considered redundant. The membership value

check can be applied to a pixel in four directions: horizontally, vertically, and diagonally in two possible directions.

Table 5.1 shows the percentage of pixels whose membership values fall between those of their neighbouring pixels. The table contains a comparison of several check criteria, starting with the horizontally adjacent neighbours, followed by the vertically adjacent neighbours, a combination of both horizontal and vertical neighbours, diagonally adjacent neighbours in both directions, and finally a combination of all four possible directions including the two diagonals. The table was obtained by applying the appropriate check to each pixel in the membership raster, except pixels on the edge of the image. The membership raster used in these tests represents "pine" in the Purau image, and was produced using the AFCM algorithm with $m = 1.5$, as discussed in the previous chapter.

Direction	Average % of redundant points
Horizontal	80.06
Vertical	80.73
Horizontal and vertical	69.26
Diagonal	65.04
All four directions	57.23

Table 5.1: Comparison of level of redundancy detected by the in-betweening method using different checks.

The results in the table show that as the number of checks is increased the percentage of points that can be removed decreases. This is because there is a greater chance of a pixel failing one of those checks. The values in Table 5.1 change with the characteristics of each membership raster. However, the relative placings of each check criterion should remain constant. If the raster contains sharp changes in membership values from one pixel to the next, then the number of removable pixels will be low, as the membership values of many pixels will fall outside the range defined by their neighbour's membership values. Alternatively, if the raster exhibits little variation then it can be represented by a TIN using very few points.

5.3.2 Averaging

Averaging is similar to the in-betweening process, but has been refined on the assumption that most processing on the resulting TIN will use linear interpolation to determine membership values for points which are not vertices of a triangle. Rather than simply checking that a pixel's membership value is between the membership values of the pixel's

neighbours, the average or half-way point between the membership values of the two neighbouring pixels is calculated and compared to the value of the central pixel. If the value of this pixel is within a pre-specified range from the average value, then the pixel contains little information and can be considered a redundant point. If the value of the pixel is outside the acceptable range, then the pixel is added to the set of points that are used to form the TIN.

A routine similar to that for the in-betweening process was performed for the averaging method. The only changes necessary were the modification of the check decision criteria used in the in-betweening method. The results of this test are presented in Table 5.2. A pre-specified range of 10 was used in the experiments to produce these results. A range of 10 was found to classify over 50% of pixels as removable, yet still produce a close approximation to the original membership raster.

Direction	Average % of removable points
Horizontal	81.29
Vertical	80.54
Horizontal and vertical	75.06
Both Diagonals	70.79
All four direction	69.27

Table 5.2: Comparison of level of redundancy detected by the average method using different checks.

A possible variation on the averaging approach is to calculate the maximum tolerance as a percentage of the difference between the membership values of the adjacent pixels. If the left pixel has a membership value of 200 and the right pixel has a value of 50 for example, the middle pixel will have an estimated value of 125. If the tolerance is set at 10 and the middle pixel has a value between 115 and 135, then the difference is not critical and the point can be removed. If the tolerance is set at 10 percent of the difference, then the pixel would be non-critical if its value is in the range 110 - 140.

Calculating the range of non-critical membership values as a percentage of the range defined by the neighbouring pixels may increase the range of non-critical membership values, thus producing fewer critical points. Given this, increasing the range may result in a TIN that provides a bad approximation of the membership surface. For example, if the difference between membership values for neighbouring pixels is great, so is the possible range for non-critical membership values. A linear interpolation of the membership value for the middle pixel will be the average of the neighbours. Extending the range of allowable non-critical points results in a greater potential for error.

If an absolute range is used, the question arises of what level of tolerance should be used. Figure 5.3 shows, for three different membership rasters, the percentage of redundant points in the membership raster with tolerance levels of between 0 and 50. Information about the amount of variation present in each raster can be obtained from these curves. A steep initial section of the curve and a high starting point indicates a low average level of variation over the area covered by the checking matrix. A lower starting point and flatter curve indicates greater variation in the raster. The lower line in Figure 5.3 illustrates such a case.

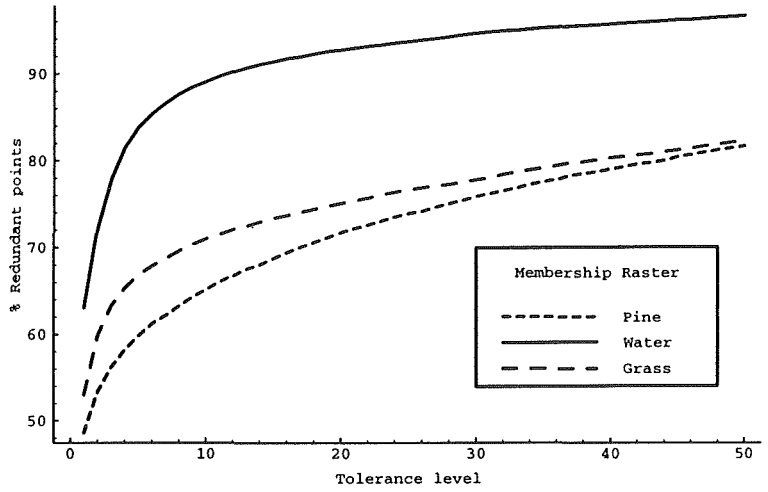


Figure 5.3: Percentage reduction over membership range for three membership rasters.

The points that remain after removing redundant points can be processed to form a TIN. The process by which a TIN is constructed from critical points is discussed in the next chapter. Each TIN is an approximation of the "surface" formed by a membership raster. The accuracy of the approximation depends on how many points were removed and where they were located in the initial raster.

A methodology was developed for the purposes of this thesis to test how various reduction schemes perform. If the AFCM algorithm is used to classify an image into three classes, three membership rasters are produced. If the three membership values are summed for any pixel, the result will be 255. When the membership values for each pixel are summed, the result will be a raster of pixels, all of which have a value of 255.

A similar approach can be used to measure a TIN's approximation of a membership raster. When the TINs produced for each membership raster are merged they should result in a flat surface with a height of 255. Two TINs are merged by interpolating a

height from the second TIN, for each vertex in the first TIN. The process is then reversed, and heights are interpolated from the first TIN for each vertex in the second TIN. Each vertex therefore has two height values, one for each TIN. The height values are summed for each vertex, and all the vertices are then combined into one set and re-triangulated.

For example, if a given point in the original image is a critical point in the membership raster for class A but is redundant in the raster for class B, the point will be a vertex in the TIN for class A but will not be part of the TIN for class B. Membership values for class B must therefore be interpolated from the class B TIN at the location of the point. Both membership values for the point are then summed. This process is performed for every point in each TIN before all the points are combined and re-triangulated. The process of merging several TINs together is explained further in Chapter 6.

By running the averaging method with various levels of tolerance, the level of error at each tolerance level can be quantified. The averaging method was applied to all three membership rasters at each tolerance level between 0 and 50. The resulting critical points from each raster were used to form three TINs. The TINs were then merged together and the level of membership for each vertex in the TIN was output. For each of the test rasters, Figure 5.4 shows the minimum, average, and maximum membership values for the resulting TIN at each level of tolerance. As the level of tolerance increases, so does the level of error as the resulting surface becomes more irregular.

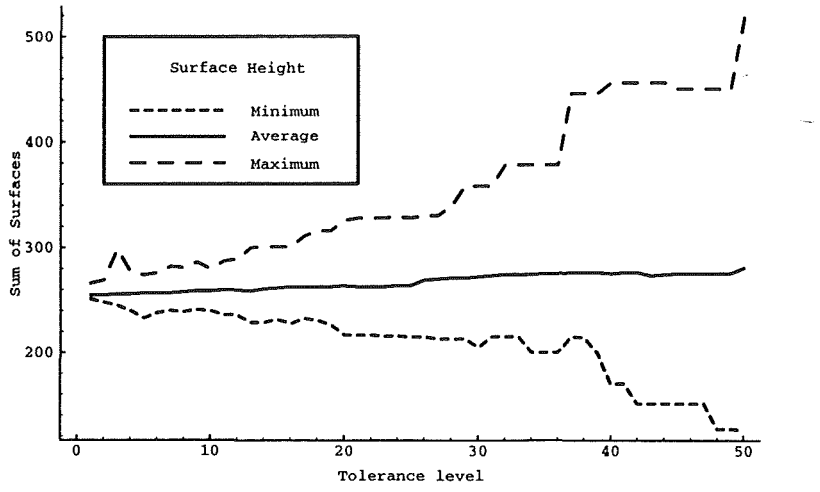


Figure 5.4: Amount of error over tolerance range when resulting TINS are added together.

In effect, a different method from those used in information theory has been used for calculating the amount of information that a pixel's membership value would contribute

to a TIN data structure. Identifying critical points using probabilities is not appropriate, as such methods do not take into consideration the characteristics of a TIN, or the subsequent processing of a TIN.

The difference is that those text compression methods that calculate the entropy of a character in a string use only the preceding characters to form the context. Strings are usually compressed or *encoded* sequentially, starting at the first character. The first character to be encoded often has no context, thus when the string is de-compressed, the first character can be decoded without using a context. The first character then becomes the context for the second character, and so on until the entire string is decoded. A character cannot be de-compressed until its context is known. Therefore a context can only include the character positions that have already been decoded.

When identifying critical points for encoding a TIN, the prediction of a pixel's membership value in order to compress the data is not of interest, rather, the point of interest is the difference between the predicted membership value and the actual membership value of each pixel. For this reason, the context for a pixel can include any of the pixel's neighbours.

5.4 Summary

For satellite images of areas covered by many small clusters of different vegetation type, the membership images produced by the FCM algorithm are difficult to reduce, as there is a large amount of variation over the membership rasters. For rasters that contain a high level of variation, it may be necessary to leave the data in a raster form and apply no data reduction techniques. Many operations are available for processing rasters, including equivalents for the TIN operations to be discussed in the next chapter, and possibly others that are not available for the TIN data structure. As the number of vertices in a TIN increases, the overhead needed to maintain network topology will reduce the benefits of reduced storage requirements.

If a membership raster exhibits significant variation from one pixel to the next over the entire raster, there are two possible explanations: there is a large amount of variation in the actual environment, with sharp changes from one class to another, or the variation is a result of inadequate resolution in the original image. The variation of physical features such as the distribution of individual trees, in the area covered by a single pixel is averaged by the sensor in the satellite (Pathirana 1992). Therefore a low resolution sensor will average a lot of information, losing the detail that would be useful for recording

variation. If the finer detail, revealed by using an image of higher resolution, shows that there is a gradual change in membership value over several pixels, the detail may be discarded, resulting in a collection of critical points similar to that obtained from a coarse image.

Using entropy to determine which pixels should be considered critical was found not to be as effective as the average check. Problems arise with the former when dealing with rasters containing many small areas that belong to one class, and large areas of intermediate membership values. Because entropy is a function of the probability of a pixel having some value, the entropy of a pixel representing one small area of a given class may be very high, as such areas are out-numbered by the pixels that constitute the boundary regions. The entropy of a pixel in the boundary region, between the many small areas of maximum membership value, may be very low even though the pixel is in a region of greater variation.

The number of pixels that can be removed without losing information, using the techniques described, is primarily dependent on the amount and direction of variation from one pixel to the next. If membership values increase or decrease slowly over the distance of several pixels, the methods described in this chapter identify many pixels as being redundant. Membership rasters containing sharp changes in value may be represented more effectively using different methods. However, other methods may require the data to have other properties such as a regular pattern of spatial distribution. Chapter 6 discusses other approaches to representing a membership raster.

The method used to reduce the amount of data will affect results from later processing. Analysts should be aware of the effects of the various methods, and any consequences for further processing. Linear interpolation is a fast and efficient method for obtaining values for intermediate locations on a triangulated surface. It has a disadvantage in that each side of a triangle represents a discrete change in slope (Watson 1992). Because linear interpolation is a popular "first off" method for surface manipulation, and the averaging method produces good results based on this fact, the averaging method appears to be the most useful.

Chapter 6

Storing and Processing Critical Information

The previous chapter described methods for removing redundant information from a membership raster so that the remaining pixels could be used to produce a triangular irregular network (TIN). This chapter is concerned with the reasons for using a TIN data structure to represent membership rasters, and other issues relating to the TIN data structure, such as triangulation algorithms, indexing, merging, and contouring. A description is given of a method for combining several TINs, similar to the overlaying of polygons or multiple rasters, so that the value at a given point is the sum of the values for the corresponding point on all the contributing TINs.

6.1 Surface modelling

The membership rasters produced by the fuzzy c-means (FCM) algorithm described in Chapter 4 are similar to rasters of other single-variable continuous field data, such as temperature, elevation, and ozone layer densities. This type of data can be represented using several quite different approaches, including:

- mathematical approximation using high order polynomials,
- rasters, and
- point interpolation.

The best approach often depends on the nature of the data. Methods used to represent terrain differ significantly from those used to represent ozone layer density. Certain

characteristics of a feature being measured may not be representable using some models. For example, a sharp change in elevation at a cliff face is difficult to represent using a polynomial, while ozone levels in the stratosphere change very gradually over space and can therefore be represented very efficiently by mathematical approximation using very few parameters.

6.1.1 Polynomials

High order polynomials and Fourier analysis have been used extensively by engineers for compressing both one and two dimensional waveforms. A classic example is the use of Fourier analysis to construct high order polynomials using combinations of sine and cosine functions to approximate analogue audio waves. Calculating polynomials, and Fourier analysis, are computationally intensive. For this reason, specialised digital signal processing (DSP) chips have been developed to perform the calculations in hardware rather than in software.

Polynomial approximations are suited to applications where the data being approximated follows a general pattern. If data contains a large amount of irregular variation, approximations using polynomials may produce additional variation due to the unpredictable oscillations of the polynomials (Petrie 1990). Because high order polynomials are designed for use with continuously varying surfaces, they provide no easy method for incorporating features such as break-lines, boundaries and holes in the representation. Terrain features such as cliffs and lakes are therefore difficult to represent using a polynomial approximation. Several polynomial patches could be combined in one surface, with the joins between patches forming sharp changes in slope.

6.1.2 Rasters

The raster model has traditionally been the data structure of first choice for representing information collected over a continuous area. The popularity of the raster model also has benefited from the integration of satellite imagery and remote sensing into GIS.

Because a raster is composed of discrete pixels, it is difficult to combine several data sets from different sources unless the pixel sizes are consistent. Hence it may be an inefficient representation of continuous change at any resolution. The problem of inconsistent pixel sizes can be solved by re-sampling each raster using a common pixel size. Re-sampling involves converting a raster from one resolution to another. The conversion averages pixel values if the resolution of the final raster is lower than that

for the initial raster, or interpolates new values if the resolution is to be increased. Resampling in this manner can introduce errors, and result in misinterpretation of the level of accuracy of the data. The size of each pixel must be small enough to accurately represent the variation across the surface, resulting in redundant pixels for large areas of similar value (Milne 1992). Given that a spatial pattern exists, for example, one tree surrounded by an area of grass, Carter (1988) states that the spatial pattern, in this case the tree, will be lost in a raster if the pixel size is greater than half the period of the spatial pattern.

Second generation rasters are often produced by classifying pixels in a satellite image, or from field data collected from sample points in a regular grid. A raster can also be produced from irregularly sampled data by interpolating values for each pixel from the values of neighbouring points. One commonly used method for performing the interpolation is known as Kriging (Burrough 1986). Once a raster has been interpolated from the original sample points, those original points are discarded. The spatial distribution of the pixels in the raster bear no relationship to the distribution of the original sample points collected from the field. Original sample points may be densely clustered in areas where there is evidence of rapid or irregular change, and sparsely distributed where little change is expected. Because the raster model is composed of a regular grid of pixels, the distribution of pixels is dependent solely on the resolution of the raster.

6.1.3 Point interpolation

Triangular irregular networks, when used for digital elevation models (DEM), are included in the family of point-interpolated surfaces, for which a height for any location on the surface can be interpolated from known heights of surrounding points. Each vertex in a triangulation corresponds to a sample point in the field or to a vertex on a contour line. The process of triangulating the points provides a framework for identifying which of the vertices of the triangulation should be used to interpolate values for any unknown point. In the case of a linear interpolation, the triangle in which a point falls is found to determine the vertices and their elevations, from which an elevation for the point can be interpolated. Interpolation methods are discussed further in Section 6.2.

The principal advantage of using a triangulation approach is that the location of triangle vertices can be controlled to minimise the number of triangles needed to give a good approximation of the surface. Where there is little variation in the surface, few points are required to represent the changes. In areas where a surface exhibits greater variation, more points can be sampled to capture this variation. By using a TIN, the necessary detail can be captured in some areas, while other areas remain sparsely covered by vertices.

This avoids having to store large amounts of information for an entire region when a large amount of detail is required for only one small area of the region.

Surface representations based on triangles can use every sample point directly, without performing any re-sampling to produce regular grids or patterns. In some cases, using every sample point is not desirable; membership rasters are one such case as they may contain redundant information. In these situations, techniques similar to those described in Chapter 5 can be used to determine which of the sample points should be used to construct triangles. Other methods for surface representation, a raster for example, often discard the original data points after the new raster representation has been generated (Petrie 1990, Milne 1992). In this way, the spatial distribution of the triangle vertices reflects the characteristics of the surface being represented better than the regular pattern of a raster.

The previous chapter described methods for removing redundant pixels from a membership raster, leaving an irregular collection of pixels. These methods were designed so that the set of pixels (or critical points) selected should provide a good approximation of the original membership raster from which the pixels were taken. Critical points were identified by measuring the difference between the membership value of a pixel and the average membership value for the surrounding pixels. Many points are identified as critical where the surface is rough and there is a difference between the average membership value and the central pixel's value. Fewer points are identified where the membership value of a pixel is the same or similar to the average value of its neighbours.

A large amount of field data such as soil nutrient levels may also be collected as point samples. Such samples are usually of natural features that exhibit continuous change. These samples are often processed into polygon or raster representations. Many features cannot be measured using satellite remote sensing; some examples are subterranean features, and invisible characteristics such as temperature and air pressure. These phenomena are valuable for many types of analysis, from crop growth potential to erosion analysis, and can easily be represented over a continuous space using a TIN.

By representing continuous information using triangulations of critical points or point samples, and accepting the overhead of constructing and maintaining a TIN, data collected at different resolutions and scales can be easily incorporated into one set, without any modification to the original values. The problems associated with the raster model, such as re-sampling data to common resolutions, are avoided when using TINs. Provided that the sample locations are specified using a common spatial coordinate system, a requirement also of the raster model, points from two separate field surveys can be

combined by triangulating them in one TIN. Certain restrictions remain, such as that the data should have been collected within a time frame that would have prevented any significant changes in conditions occurring between the collection of sample values.

When representing information that changes continuously over an area, a TIN can be considered half way between a raster representation and a vector based representation. Raster models consisting of a single, regular grid of pixels or cells can be divided into two types: those where each pixel contains a value quantifying some characteristic or feature; and those where the value of a pixel reflects which of several characteristics or features is present at that location. DEMs and membership rasters fall into the former type, while the results generated from a traditional classification are of the second type. The TIN representation provides a means by which the first type of raster can be converted to a polygon representation.

DEMs and membership rasters can easily be converted to TINs via critical point identification, and TINs can be converted to rasters by interpolation. A raster containing pixels, each of one of several classes, can be converted to a polygon representation directly. TINs can be converted to a vector representation, and vectors to TINs, via contouring and triangulation. Converting from polygons to TINs is difficult and may result in significant distortions of the data. Because a polygon structure represents data in a processed state where information about variation over space is not available, the conversion is not likely to produce accurate results.

The following sections in this chapter discuss reasons for and against using TINs to represent the surfaces produced by the fuzzy c-means classification algorithm. Topics discussed include:

- generation of TINs,
- issues of storage efficiency,
- combining of multiple data sources, and
- functions that can be applied to TINs.

6.2 Delaunay triangulation

There are several approaches to creating a triangulation from a collection of points. Early triangulation methods were slow, and often resulted in different triangulations depending on which point was selected as a start position (Petrie 1990). When building a triangulation there are two main objectives: to build triangles that are as equilateral as possible;

and to minimise the length of each side of the triangles. Algorithms that achieve these two objectives avoid producing triangulations that contain elongated triangles. Such triangles can produce inaccurate results when values are interpolated from the vertices. The Delaunay triangulation of a collection of points fulfills these two requirements.

Delaunay triangulation is associated with Thiessen polygons. Given a set of vertices, Thiessen polygons define the region influenced by each vertex. A polygon is constructed around a vertex such that any point within the polygon is closer to the central vertex than any other vertex outside the polygon.

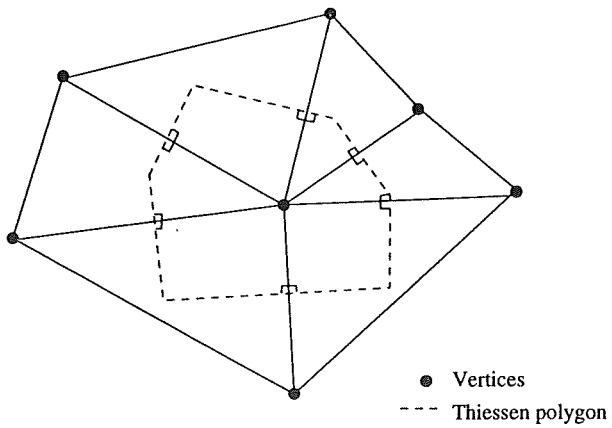


Figure 6.1: Thiessen polygon for a vertex.

6.3 Triangulation efficiency

Although triangular structures are an efficient means of storing irregular gridded data, processing of these structures can be inefficient due to the overhead of maintaining the topological relationships of the triangular network. A TIN is a much more complex data structure than a raster. Triangles must be created by linking vertices together in such a way that edges do not cross, and all vertices have at least two connecting edges. These topological conditions must be maintained when points are inserted and removed from the TIN. Implementing a data structure to represent this topology introduces a storage space overhead for each point in the triangulation. The task of maintaining this structure also adds to the processing time for TIN operations.

The exact overheads involved with preserving a topological structure depend on the data structure used for storing the links between vertices. Some suitable data structures include *doubly connected edge lists*, as described by Preparata & Shamos (1985) and used

by De Floriani & Puppo (1992), and the *triangle array* method as used by Hayton (1992). The number of vertices in a TIN may reach some level such that the cost of extra memory requirements and processing time cancel out the effect of the reduced number of points required for a TIN to represent membership values for an area.

6.4 Indexing triangular structures

Indexing irregular triangular structures is difficult because the shapes formed by triangles do not follow a regular pattern. When some form of area indexing scheme is introduced, such as quad-trees, there will be some overlap of triangles from one quadrangle to the next. It is usually impossible to build a series of non-overlapping quads such that each triangle is completely included within a single quadrangle, unless the triangles are constructed from vertices sampled on a regular grid. Similar problems are also encountered when using irregular shapes for the index (De Floriani 1989).

Possible solutions have been offered, all of which have advantages and disadvantages. The following three sections describe several methods that can be categorised into one of the following classes:

- uniform grid, range search,
- irregular triangles,
- guided search.

6.4.1 Uniform grids and range searches

Use of uniform grids and range searches involves placing an imaginary grid over the spatial domain of the data set. A list is maintained, for each grid cell, of points and triangles that fall within or overlap that cell. The grid cells form an index into the entire data set. When new points are to be added to the triangulation, the cell containing the location of the point can be found quickly, and the number of triangles to search is significantly reduced.

Alternatively, the grid lists can be built prior to the triangulation process. When triangulation begins, the grid can be used to locate points that are close together. This is useful when forming triangles, as the grid lists significantly reduce the number of points to be searched (Huang 1989, Fang & Piegl 1993). The effectiveness of using a regular grid depends on the spatial distribution of points within the TIN. If certain parts of the TIN

are densely populated with points, and other parts of the TIN remain sparsely populated, some grid cells will have a very long list of associated points while other cells will have a very short list of points. Hence, where there is high local variation, this method will have significant overhead.

6.4.2 Irregular triangles

The problem of irregularly distributed data points can be addressed by using irregular shapes to index triangular structures. Indexing structures can be balanced so that there is approximately the same number of points contained in each cell of the index. In this way, the search time to find any point is constant, and independent of the spatial distribution of points. Such indexing schemes require extra processing when points are inserted and deleted from the TIN, as the underlying indexing structure must be updated and balanced with each insertion or deletion. The *Cell Tree* described by Gunther (1988) is a good example of a spatial index that uses irregular shapes. With a quadtree, an area is recursively subdivided into smaller and smaller squares but, for a cell tree, irregular shapes are used for each cell.

De Floriani (1989) described a slightly different method based on recursively indexing one triangulation with another triangulation that contains fewer points. This indexing scheme was later used by De Floriani & Puppo (1992) to increase the speed of an on-line algorithm, where the triangulation could be built without spatially sorting vertices before processing. Two different criteria can be used to select which points from the detailed TIN should be used to form the indexing TIN: spatial distribution or surface accuracy. The indexing TIN can be created in such a way that the distribution of points within the index can be evenly spread over all indexing triangles. Alternatively, points from the original triangulation can be chosen in such a way that the indexing TIN, although containing fewer points, can be used to produce an acceptable approximation of the original TIN.

6.4.3 Oriented-walk search

The oriented-walk search, described by Hayton (1992), requires no indexing data structures but does not adapt to the spatial distribution of points within a TIN. The search traverses a TIN, checking each triangle to see if it contains the point being inserted or deleted. The algorithm traverses the TIN in a way that significantly reduces the number of triangles that must be searched.

When a triangle containing some point must be found, the oriented walk is begun at an arbitrary triangle. The edges of the triangle are traversed in a clockwise direction. If

the point is located to the right of all three edges, then the triangle must contain the point. If, during this test, the point is found to lie to the left of an edge, the algorithm moves to begin searching the triangle adjacent to the edge that failed the test. The algorithm can be improved if we can assume that points are likely to be inserted in some known spatial order. Each new search can then be started where the previous search finished. If this assumption is valid, search times can be greatly reduced.

6.5 Interpolation

Interpolation is the process by which an estimate of the value of a function at some location is obtained from known values of the function at surrounding points (Watson 1992). A surface may be described by a function, where the surface value is a function of the (x,y) location. For example, in the case of a function representing elevations across a field, the height of some given point can be estimated by using the function representing the surface and the heights of nearby points to calculate a suitable value. There are many different approaches to interpolation, but all aim to produce estimates that minimise the difference between the interpolated value and the actual surface.

Each interpolation method is based on some interpretation of how the known value of each vertex influences the unknown values in the surrounding region. Once the influences are determined, it is possible to estimate values for the unknown locations from the vertices. Watson (1992) provides a survey of many different approaches to surface interpolation, including several that are specific to irregular triangulations. Methods for triangulation can be divided into two distinct types: linear, and nonlinear. Linear methods are simple to implement and fast to execute, and produce surfaces within which each triangle is a plane with constant slope and aspect and triangle edges represent a distinct change in slope. For these reasons, linear interpolation is possibly the most commonly used approach when interpolating from triangulated data.

The non-linear interpolation methods produce smoother surfaces than does linear interpolation. Non-linear methods produce surfaces for which there is a gradual change in slope from one triangle to the next. These methods usually incorporate an estimate of the surface gradient at each data point into the interpolation calculation. By incorporating information on gradient, the surface within a triangle can be non-planar, thus allowing a smooth change from one triangle to the next.

6.6 Resolution independence

Points can be added to a triangulation to increase resolution and coverage over areas of particular interest. If there are areas for which it is important to have closely spaced sample points, these can be added to a TIN by incorporating extra points into the triangulation. To perform a similar task using a single raster is difficult. Nesting one raster inside another, in a similar manner to quadtrees, could provide an alternative method. However, maintaining a structure such as this also creates overheads, and unless irregular shaped rasters are supported there will still be some amount of redundancy. Because a raster contains a collection of identically sized pixels, to increase the density of samples over a small area would involve increasing the density over the entire raster. By re-sampling a raster at a higher resolution, areas that have small changes in membership value will be represented by a greater number of pixels. The changes from one pixel to the next may be so insignificant as to cause the extra pixels to be redundant.

Because the location of points within a TIN can be stored with arbitrary precision, data from various sources can be incorporated into the same triangulation, the only requirement being that the data is spatially referenced using a common projection. Collections of sample points from several surveys of the same characteristic can be combined to form a single data set. Alternatively, several collections of sample points with values for different characteristics can be processed simultaneously to produce a resulting triangulation that is a function of one or more of the characteristics being processed. The processing of several TINs representing different characteristics is discussed in the following section.

6.7 Merging triangulations

The overlay of several TINs is similar to forming the union of several sets of polygons. For a raster, there is grid arithmetic. We can add values in one grid to values in another, for example. Other grid operations can be done, such as subtraction, multiplication, or division. TIN overlay provides a method for performing the same functions for TINs. The process of adding two TINs should result in a surface that is the sum of values from each contributing surface. For example, if a given point in one triangulation has a value of 100 and the value at the same location in another triangulation is 75, then the corresponding point in the resulting TIN should have a value of 175.

Determining the best area in which to grow a particular crop provides an example of how operations on TINs can be useful. The growth rate of some crop may be dependent on three main variables: soil moisture, temperature, and nutrient levels. Suppose that

each of these characteristics is measured over the study area, and the values normalised to fall in the range 0 . . . 255. Fuzzy classification is one possible method that could be used to obtain such measurements. Each characteristic plays an important role in the growth rate of the crop, although some characteristics will be more important than others. For this reason each characteristic is assigned a weighting factor between 0 and 10, as shown in Table 6.1, that reflects their importance for a given crop.

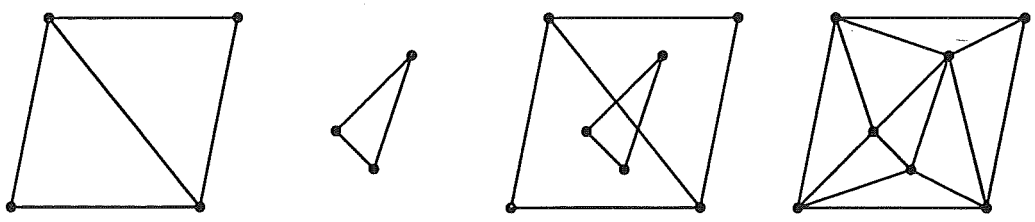
Characteristic	Weight
Soil moisture	7
Temperature	6
Nutrient	9

Table 6.1: Weighting factors for each characteristic.

Each surface can be multiplied by its weighting factor and then added to the other surface. Peaks in the resulting surface represent areas that are most suitable for growing that particular crop, while troughs represent areas that are unsuitable.

Overlaying of multiple TINs is relatively simple, and can be split into three steps:

- interpolate corresponding values,
- perform the mathematical operation, and
- re-triangulate the points.



(a) TIN A (b) TIN B (c) Overlay of A and B (d) Triangulation of the overlaid TINs

Figure 6.2: Merging two triangulations.

The process of merging two TINs is shown in Figure 6.2. The first step is to take each point in TIN A and to interpolate a value for the same location in TIN B. The same

interpolation operation is performed for TIN B, but values are interpolated from the first TIN (A) instead. In some cases, vertices from one TIN will fall outside the *convex hull* of another TIN. The convex hull of a TIN is the polygon formed by line segments connecting all the outer vertices of the TIN, and defines the region for which points can be accurately interpolated. When points fall outside the convex hull, the value for the point is extrapolated from the value of the nearest neighbours. After the interpolation process, each point has two values, one for each surface. These two values can be added, subtracted, divided, or multiplied as the user specifies, to produce a final value. Once the the total value for each point has been calculated, the points can be re-triangulated.

In a series of experiments undertaken on combining multiple TINs into one, all the TINs were generated from membership rasters produced from the same satellite image. Hence, each vertex originated from the same regular grid and where one characteristic changed, it was likely that others would also change. This is because characteristics such as soil type and vegetation are often correlated with each other, or with another characteristic such as the type of terrain. For these reasons, when three TINs (representing membership levels in: pine, water, and grass) produced by identifying critical points in the membership rasters generated by the FCM algorithm were merged, there was a very high degree of overlap in the sets of points that make up each TIN.

The original three TINs were generated from critical points identified using the averaging method described in Section 5.3.2, with the tolerance levels ranging from 1 to 50. Figure 6.3 shows how the number of points in the resulting TIN increased as the tolerance level was increased. The curve shows the percentage increase in the number of vertices from the largest of the source TINs to the resultant TIN. At a tolerance level of 10, the TIN produced by merging the three TINs contains approximately 3% more vertices than the largest source TIN. At low levels of tolerance, very few points in the membership rasters from which the TINs were generated are redundant. Therefore each TIN contains a large number of points. These points are usually located in different positions in each TIN. Hence, there is a greater number of unique points in each TIN, which explains the high level of increase for low levels of tolerance.

Figure 6.4 shows how membership surfaces representing two separate hypothetical classes can be merged, with membership values summed, to produce a third surface.

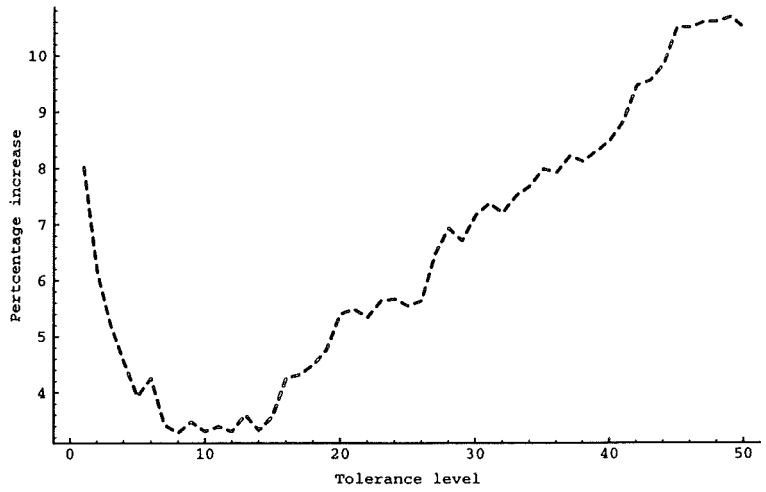


Figure 6.3: Percentage increase in resulting TIN size.

6.8 Contouring

Section 6.1.3 described the TIN model as being between the raster model and the polygon model. Chapter 5 described methods by which data can be extracted from a raster and used to construct a TIN. We now describe how a fuzzy set operator, the alpha cut, can be used to produce polygons from a membership surface.

The alpha cut operator was described in Section 3.1.2, as applied to a one-dimensional membership function. The alpha cut is the process of intersecting a membership function at a specified membership level, and is therefore equivalent to drawing a contour line on a membership surface. An alpha cut can therefore be used to produce polygons from a membership surface. Each polygon will contain points that have at least the same level of membership as the cut level.

Other features of the TIN model that are used when representing terrain can also be useful for TIN models that represent membership values. Break-lines and boundaries are often used in terrain models to accurately represent ridge lines, gullies, lakes and coast lines. A TIN representing class membership values may also have to include lines at places where there are sudden changes in membership value. Such changes can occur at features such as roads or shorelines.

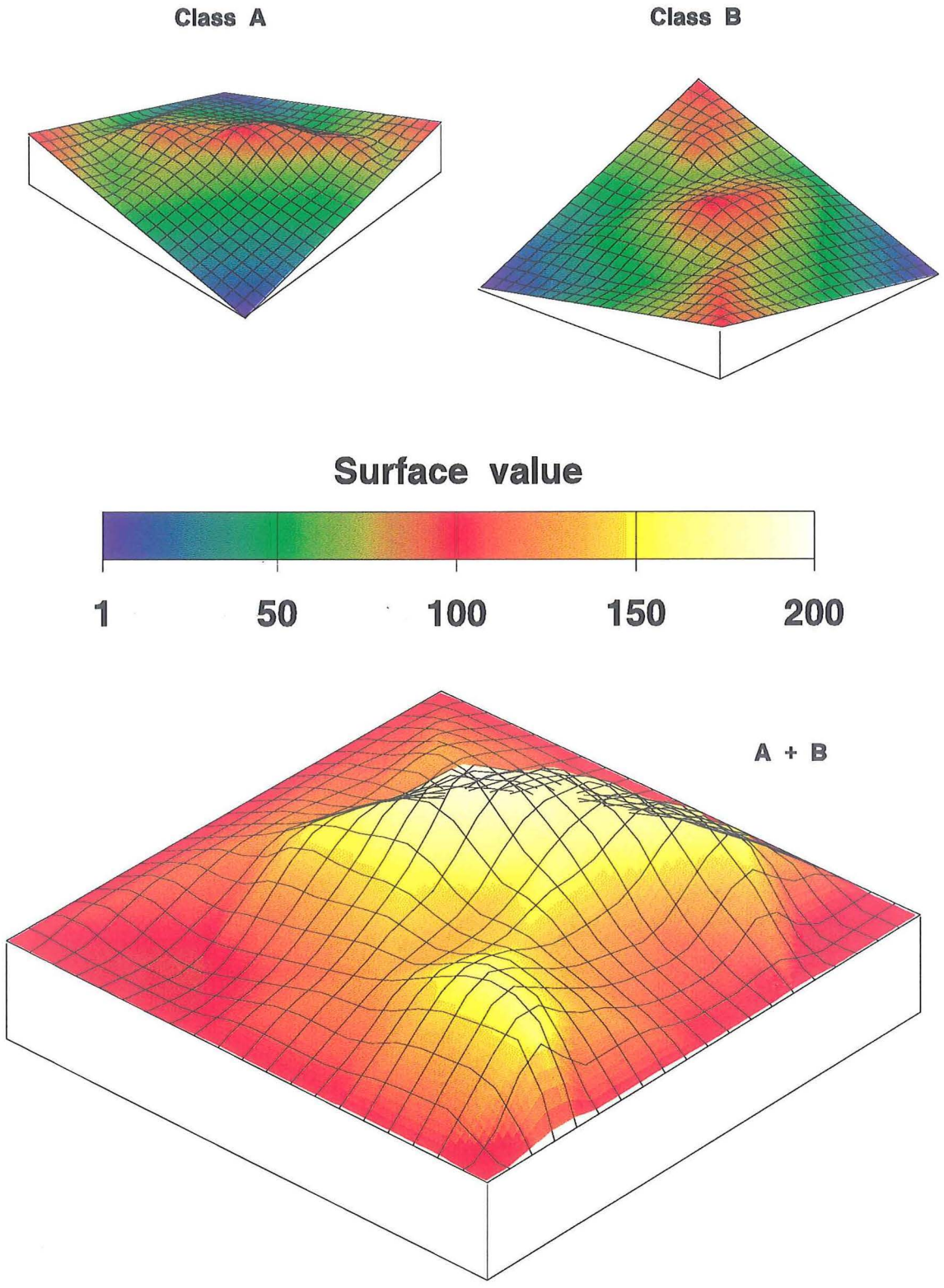


Figure 6.4: "Adding" two membership TINs together.

6.9 Summary

Information that is measured on a continuous scale and over a continuous area can be represented efficiently, and to a reasonable level of accuracy using a TIN data structure, provided that the amount of variation from one sample point to the next is small. The TIN model provides a flexible framework for combining data from several sources, avoiding the problems associated with re-sampling. Although other methods, such as the use of high order polynomials or the raster model may allow data to be processed faster, the models may not easily produce an accurate approximation of the data being represented. The majority of functions available for processing other models can also be implemented for a TIN. Because many of the characteristics being measured are dependent on terrain and surface topography, using a data structure similar to that used to represent terrain is appealing.

Chapter 7

Conclusions

Most land classification methods are based on some form of Boolean classification theory. Any area is in one of two possible states with respect to a land use class; either it is a member of a particular class, or not a member of that class. Classifying an area in this way removes information about the way in which class membership varies across the spatial domain of interest. Incorporating fuzzy logic into the process of classifying pixels in a remote sensed image can retain information that is useful for representing geographic features with indistinct or fuzzy boundaries.

Calculating class membership on a continuous scale significantly reduces the amount of classification error when compared to the Boolean membership values produced by earlier classification methods. By calculating membership values in each class for each pixel, a high level of information can be represented describing the transition zone between two regions.

A fuzzy classification algorithm produces a membership raster for each class in the classification. Each pixel in such a raster contains a membership value indicating the level of membership for that pixel in the class represented by the raster. This results in a large amount of data having to be stored for later processing.

Membership rasters are similar to raster-based digital elevation models in that each pixel has a single value, in this case a membership value rather than a height as in the case of a DEM raster. In the same way that TINs are used to represent DEMs, TINs can be used to represent membership rasters. Representing membership rasters in this way reduces the amount of storage space required, and allows algorithms developed for the TIN model to be used for processing data about continuously varying features.

A practical example has been given in this thesis, showing how a multi-spectral satellite image can be classified using the fuzzy c-means algorithm to produce membership

values for each pixel in each class. Membership rasters produced by the FCM algorithm were processed using four different methods to identify critical points in the data set. Two of the methods described use information theory and context modelling to identify critical points, while simple in-betweening and an averaging process were used in the final two methods. Essentially, critical points were chosen by calculating their usefulness in constructing a TIN to represent the membership raster to some desired level of accuracy. Each critical point is used as a vertex in the TIN. Identifying and removing non-critical points can result in an 80% reduction in the number of points needed to represent a membership raster to some desired level of accuracy. The exact amount of reduction that can be achieved does however depend on how much membership values change from one pixel to the next, and the level of tolerance specified by the user. Together, these two variables influence how much information is lost in the reduction process.

An example was also given of how multiple TINS can be overlaid and the surface values summed to form a new TIN. Overlaying is not an operation normally performed on TINs that represent terrain. Overlaying such TINs has little practical use, except possibly in GISs that are designed specifically for dealing with three dimensional data, such as those systems used in mining and geological applications. However, because membership TINs are usually generated for several classes over the same area, the overlay operation becomes an important analytical tool. Overlaying membership TINs is similar to overlaying lines and polygons in a vector-based model where attribute values from each of the features being overlaid are combined for use in later database queries.

The overlay operation was used to measure the accuracy of surface approximations constructed from critical points identified using the four methods described in this thesis. By overlaying the membership TINs for three classes over the same area, the resulting surface should be flat, and have a total membership level of approximately 1.0. By varying the methods used to identify critical points, and performing this test on the resulting TINs, the accuracy of each method was compared. It was found that the best method for identifying critical points was to identify those pixels with membership values that were not similar to the average membership value of surrounding pixels.

Clearly, it is a good idea to represent features that vary continuously over a spatial domain by using fuzzy logic to calculate membership values. The likelihood of classification error or source error being introduced into the data is very low. The principal problem of using a membership raster for every class is the high storage space requirements. This research has shown that storage requirements can be significantly reduced by using a TIN data structure to replace each membership raster.

Bibliography

Bell, T. C., Cleary, J. G. & Witten, I. H. (1990), *Text Compression*, Prentice Hall.

Bezdek, J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press.

Bezdek, J. C., ed. (1987), *Analysis of Fuzzy Information*, CRC Press.

Bezdek, J. C., Ehrlich, R. & Full, W. (1984), 'FCM: The Fuzzy C-Means clustering algorithm', *Computers and Geosciences* **10**(2-3), 191–203.

Burrough, P. A. (1984), 'The application of fractal ideas to geophysical phenomena', *The institute of mathematics and its applications* **20**(3/4), 36–42.

Burrough, P. A. (1986), *Principles of Geographical Information Systems for Land Resources Assessment*, Clarendon Press, Oxford.

Cannon, R. L., Jitendra, D. V. & Bezdek, J. C. (1987), 'An approximate Fuzzy C-Means alogorithm', *Analysis of Fuzzy Information* **3**, 153–167.

Carter, J. R. (1988), 'Digital representations of topographic surfaces', *Photogrammetric Engineering and Remote Sensing* **54**(11), 1577–1580.

Cetin, H. & Levandowski, D. W. (1991), 'Interactive classification and mapping of multi-dimensional remotely sensed data using n-dimensional probability density functions (nPDF)', *Photogrammetric Engineering and Remote Sensing* **57**(12), 1579–1587.

Chen, Z.-T. & Guevara, J. A. (1987), 'Systematic selection of very important points (vip) from digital terrain model for constructing triangular irregular networks', *Auto-Carto 8* pp. 50–55.

Chrisman, N. R. (1989), Modeling error in overlaid categorical maps, *in* Goodchild & Gopal (1989), chapter 2, pp. 21–34.

Csillag, F. (1992), 'Resolution, accuracy and attributes: Approaches for environmental geographical information systems', *Computers, Environment, and Urban Systems* **16**, 289–297.

De Floriani, L. (1989), 'A pyramidal data structure for triangle-based surface description', *IEEE Computer Graphics and Applications* pp. 67–78.

De Floriani, L. & Puppo, E. (1992), 'An on-line algorithm for constrained delaunay triangulation', *CVGIP: Graphical Models and Image Processing* **54**(3), 290–300.

Delaunay, B. (1934), 'Sur la sphere vide', *Bulletin of the Academy of Sciences of the USSR, VII Classe. Sci. Mat. Nat.* pp. 793–800.

Drummond, J., Van Essen, R. & Boulerie, P. (1991), 'Some considerations on vectorizing algorithms', *ITC Journal* **3**, 153–157.

Durst, M. J. & Kunii, T. L. (1992), Methods for the efficient storage and manipulation of spatial geological data, *in* A. K. Turner, ed., 'Three', Kluwer-dimensional modeling with geoscientific information systems, chapter 14, pp. 189–214.

Erd (1991), *Erdas field manual.*

Fang, T.-P. & Piegl, L. A. (1993), 'Delaunay triangulation using a uniform grid', *IEEE Computer Graphics and Applications* pp. 36–47.

Goodchild, M. F. (1976), Statistical aspects of the polygon overlay problem, *in* 'Harvard Papers on GIS', Vol. 6, Publisher unknown.

Goodchild, M. F. (1980), 'Fractals and the accuracy of geographical measures', *Mathematical Geology* **12**(2), 85–98.

Goodchild, M. F. (1989), Modeling error in objects and fields, *in* Goodchild & Gopal (1989), chapter 10, pp. 107–113.

Goodchild, M. F. & Gopal, S., eds (1989), *Accuracy of Spatial Databases*, Taylor and Francis, 4 John Street, London, WC1N 2ET.

Goodchild, M. F., Guoqing, S. & Shiren, Y. (1992), 'Developement and test of an error model for categorical data', *Internatinal Journal of Geographic Information Systems* **6**(2), 87–104.

Griffith, D. A. (1983), 'The boundary value problem in spatial statistical analysis', *Journal of Regional Science* **23**(3), 377–387.

Gunther, O. (1988), *Effiecient Structures for Geometric Data Management*, Springer-Verlag.

Haining, R., Griffith, D. & Bennett, R. (1983), 'Simulating two-dimensional autocorrelated surfaces', *Geographical Analysis* **15**(3), 247–255.

Hall, G. B., Wang, F. & Subaryono (1991), 'Comparison of boolean and fuzzy classification methods in land suitability analysis using GIS', *Environment and Planning A*.

Harris, R. (1987), *Satellite Remote Sensing An Introduction*, Routledge and Kegan Paul.

Hayton, M. (1992), An investigation of triangulation methods, Honours degree project report, Computer Science Department, University of Canterbury.

Helman, D. R. & Langdon, G. G. (1988), 'Translating data from a useful form to an economical one', *IEEE Potentials* pp. 25–28.

Heuvelink, G. B. M. & Burrough, P. A. (1993), 'Error propagation in cartographic modelling using boolean logic and continuous classification', *Internatinal Journal of Geographic Information Systems* **7**(3), 231–246.

Heuvelink, G. B. M., Burrough, P. A. & Stein, A. (1989), 'Propergation of errors in spatial modelling with GIS', *Internatinal Journal of Geographic Information Systems* **3**(4), 303–322.

Huang, Y.-p. (1989), 'Triangular irregular network generation and topographical modeling', *Computers in Industry* **12**, 203–213.

Jenson, S. K. & Domingue, J. O. (1988), 'Extracting topgraphical structure from digital elevation data for geographic information system analysis', *Photogrammetric Engineering and Remote Sensing* **54**(11), 1593–1600.

Kent, J. T. & Mardia, K. V. (1988), 'Spatial classification using fuzzy membership models', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(5), 659–671.

Klir, G. J. & Folger, T. A. (1988), *Fuzzy sets, uncertainty, and information*, Prentice Hall.

Kollias, V. J. & Voliotis, A. (1991), 'Fuzzy reasoning in the developement of geographical information systems. FRSIS: A prototype soil information system with fuzzy retrieval capabilities', *Internatinal Journal of Geographic Information Systems* **5**(2), 209–223.

Kubik, K. & Patias, P. (1988), 'Robust estimation and DEM data compression', *Australian Journal of Geodesy, Photogrammetry and Surveying* **48**, 53–67.

Kvamme, K. L. (1990), GIS algorithms and their effects on regional archaeological analysis, *in* K. M. S. Allen, S. W. Green & E. B. W. Zubrow, eds, 'Interpreting Space: GIS and archaeology', Taylor and Francis, pp. 112–125.

Lawson, C. L. (1977), Software for c1 surface interpolation, *in* J. R. Rice, ed., 'Mathematical Software III', Academic Press, pp. 161–194.

Leung, Y. (1984), 'Towards a flexible framework for regionalization', *Environment and Planning A* **16**, 1613–1632.

Leung, Y. (1987), 'On the imprecision of boundaries', *Geographical Analysis* **19**(2), 125–151.

Leung, Y., Goodchild, M. F. & Lin, C.-C. (1992), 'Visualization of fuzzy scenes and probability fields', *Fifth International Symposium on Spatial Data Handling*.

Maffini, G., Arno, M. & Bitterlich, W. (1989), Observations and comments on the generation and treatment of error in digital GIS data, *in* Goodchild & Gopal (1989), chapter 5, pp. 55–67.

Mandelbrot, B. B. (1977), *Fractals: form, chance, and dimension*, Freeman.

Mandelbrot, B. B. (1983), *The fractal geometry of nature*, Freeman.

Mark, D. M. & Csillag, F. (1989), 'The nature of boundaries on 'area-class' maps', *Cartographica* **26**(1), 65–78.

Milne, P. H. (1992), 'Digital ground modelling', *Mapping Awareness and GIS in Europe* **6**(3), 33–36.

Moffat, A. (1991), Two-level context based compression of binary images, *in* J. A. Storer & J. H. Reif, eds, 'Proceedings of Data Compression Conference', IEEE Computer Society Press, Los Alamitos, California.

Pal, S. K. & Dasgupta, A. (1992), 'Spectral fuzzy sets and soft thresholding', *Information Sciences* **65**(1-2), 65–97.

Pathirana, S. (1992), 'Detection of linear and sub-pixel phenomena using the fuzzy membership approach', *Sixth Australasian Remote Sensing Conference* **2**, 424–433.

Petrie, G. (1990), Modelling, interpolation and contouring procedures, *in* G. Petrie & T. J. M. Kennie, eds, 'Terrain modelling in surveying and civil engineering', Whittles, chapter 8, pp. 112–127.

Preparata, F. P. & Shamos, M. I. (1985), *Computational Geometry: An Introduction*, Springer-Verlag, New York.

Ramer, A. (1990), 'Concepts of fuzzy information measures on continuous domains', *International Journal of General Systems* **17**, 241–248.

Robinson, V. B. (1988), 'Some implications of fuzzy set theory applied to geographic databases', *Computers, Environment, and Urban Systems* **12**, 89–97.

Samet, H., Rosenfeld, A., Shaffer, C. A. & Webber, R. E. (1984), Use of hierarchical structures in geographical information systems, *in* 'Proceedings of the International Symposium on Spatial Data Handling', pp. 392–411.

Sui, D. Z. (1992), 'A fuzzy GIS modeling approach for urban land evaluation', *Computers, Environment, and Urban Systems* **16**, 101–115.

Todd, S., Langdon, G. G. & Rissanen, J. (1985), 'Parameter reduction and context selection for compression of gray-scale images', *IBM Journal of Research and Development* **29**(2), 188–193.

Vatti, R. R. (1992), 'A generic solution to polygon clipping', *Communications of the ACM* **35**(7), 56–63.

Vemuri, B. C. & Malladi, R. (1992), 'Surface gridding with intrinsic parameters', *Pattern Recognition Letters* **13**, 805–812.

Wang, F. (1989), A fuzzy expert system for remote sensing image analysis, *in* 'IGARSS '89 - 12th Canadian Symposium on Remote Sensing', Vol. 2, pp. 848–851.

Wang, F. (1990), 'Fuzzy supervised classification of remote sensing images', *IEEE Transactions on Geoscience and Remote Sensing* **28**(2), 194–201.

Wang, F. (1991), 'Intergrating GIS's and remote sensing image analysis systems by unifying knowledge representation schemes', *IEEE Transactions on Geoscience and Remote Sensing* **29**(4), 656–664.

Wang, F. & Hall, G. B. (1991), Fuzzy representation of geographical boundaries in GISs, To be published in IJGIS.

Wang, F., Hall, G. B. & Subaryono (1990), 'Fuzzy information representation and processing in conventional GIS software: database design and application', *Internatinal Journal of Geographic Information Systems* **4**(3), 261–283.

Watson, D. F. (1982), 'Acord: Automatic contouring of raw data', *Computers and Geosciences* 8(1), 97–101.

Watson, D. F. (1992), *CONTOURING A Guide To The Analysis And Display Of Spatial Data*, Pergamon Press.

White, D. (1976), A design for polygon overlay, *in* 'Harvard Papers on GIS', Vol. 6, Publisher unknown.

Windham, M. P. (1982), 'Cluster validity for the fuzzy c-means clustering algorithm', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **4**(4), 357–363.

Zadeh, L. A. (1965), 'Fuzzy sets', *Information and Control* **8**, 338–353.

Zhang, G. & Tulip, J. (1990), An algorithm for the avoidance of sliver polygons and clusters of points in spatial overlay, *in* K. Brassel & H. Kishimoto, eds, '4th International Symposium on Spatial Data Handling', Vol. 1, pp. 141–150.

Zhang, G. & Wilkinson, S. (1991), Simultaneous multi-layer spatial overlay, Unpublished.

Appendix A

One Dimensional Fuzzy C-Means Implementation

This appendix includes a program listing, written in the C programming language, of the author's implementation of the fuzzy c-means classification algorithm. The program listed here is used for clustering data based on a single variable. A complete implementation, capable of clustering data in any number of dimensions, was used to cluster pixels in a SPOT image.

```
#include "math.h"
#include "values.h"
#include "stdio.h"

#define MAX_N 20000
#define MAX_C 10

double data_value[MAX_N];          /* 1-dimensional data values */
double membership[MAX_N][MAX_C];   /* membership for elmt. N in set C */
double center[MAX_C];              /* center for set C */

double center_old[MAX_C];          /* original centres */

int N;                             /* number of data values */
int C;                             /* number of clusters */
```

```
/*
 *   Initialise the membership values.
 */

void init_memberships(void) {
    int n;
    int c;
    double f;
    double *p;

    p = (double *)malloc(C * sizeof(short));
    for (n = 0; n < N; n ++) {
        f = 0.0;
        for (c = 0; c < C; c ++)  {
            p[c] = (double)random() / MAXINT;
            f += p[c];
        }
        for (c = 0; c < C; c ++)
            membership[n][c] = p[c] / f;
    }
    free(p);
}


/*
 *   Dump out the table of cluster centers.
 */

void dump_centers(void) {
    int c;

    for (c = 0; c < C; c ++)
        fprintf(stdout, "center[%d] = %6.2f\n", c, center[c]);
}


/*
 *   Dump out the membership table.
 */

void dump_memberships(void) {
    int n;
    int c;
```

```
    for (n = 0; n < N; n ++) {
        fprintf(stdout, "data_value[%d] = %4.1f"
                        "mem = ", n, data_value[n]);
        for (c = 0; c < C; c++)
            fprintf(stdout, "%6.4f ", membership[n][c]);
        fprintf(stdout, "\n");
    }
}


/*
 *   Read in the centers and the data values.
 */

void read_data(char *centers, char *data) {
    FILE *c_ptr;
    FILE *d_ptr;

    /* Read the initial cluster centers */

    c_ptr = fopen(centers, "r");
    C = 0;
    while (!feof(c_ptr)) {
        fscanf(c_ptr, "%lf\n", &center[C]);
        center_old[C] = center[C];
        C ++;
    }
    fclose(c_ptr);

    /* Read the data to be clustered */

    d_ptr = fopen(data, "r");
    N = 0;
    while (!feof(d_ptr)) {
        fscanf(d_ptr, "%lf\n", &data_value[N]);
        N ++;
    }
    fclose(d_ptr);
}


/*
 *   Write value and membership.
```

```
 */

void write_data(char *out) {
    FILE *o_ptr;
    int n;
    int c;

    o_ptr = fopen(out, "w");
    for (n = 0; n < N; n++) {
        fprintf(o_ptr, "%lf ", data_value[n]);
        for (c = 0; c < C; c ++)
            fprintf(o_ptr, "%lf ", membership[n][c]);
        fprintf(o_ptr, "\n");
    }
    fclose(o_ptr);
}


/*
 *   Calculate the cluster centers based on the data values and
 *   existing membership values.  (Equation 3)
 */

void new_cluster_centers(double m) {
    double weighted;                /* weighted membership value */
    double top;
    double bot;
    int c;
    int n;

    for (c = 0; c < C; c ++) {
        top = 0.0;
        bot = 0.0;
        for (n = 0; n < N; n ++) {
            weighted = pow(membership[n][c], m);
            top += (weighted * data_value[n]);
            bot += weighted;
        }
        center[c] = top / bot;
    }
}


/*
```

```
 *   Calculate the new memberships for each data value for each
 *   cluster. (Equation 4)
 */

double new_memberships(double m) {
    double top;
    double bot;
    double sum;
    double p;
    double eps;
    double leps;
    double seps;
    int c;
    int cc;
    int n;

    p = 1.0 / (m - 1.0);
    leps = 0.0;
    seps = 0.0;
    for (c = 0; c < C; c ++) {
        for (n = 0; n < N; n ++) {
            top = data_value[n] - center[c];
            sum = 0;
            if (top != 0.0)
                for (cc = 0; cc < C; cc ++) {
                    bot = data_value[n] - center[cc];
                    if (bot == 0.0)
                        break;
                    sum += pow(pow(top / bot, 2.0), p);
                }
            else {
                sum = 1.0;
                bot = 1.0;
            }
            if (bot != 0.0) {
                eps = fabs(membership[n][c] - (1.0 / sum));
                membership[n][c] = 1.0 / sum;
            }
            else  {
                eps = fabs(membership[n][c] - 0.0);
                membership[n][c] = 0.0;
            }
            seps += eps;
```

```
            if (eps > leps)
                leps = eps;
        }
    }
    return leps;
}


void dump_stats(int r) {
    float d;
    int c;

    d = 0.0;
    for (c = 0; c < C; c++)
        d += fabs(center[c] - center_old[c]);
    fprintf(stdout, "%f %d\n", d, r);
}


int main(int argv, char **argc) {
    double m;
    double e;
    int R;
    int r;

    if (argv != 8) {
        fprintf(stderr, "Usasge: %s <weight> <iterations> <centers> "
                        "<data_in> <data_out> <m|c>\n", argc[0]);
        exit(1);
    }
    sscanf(argc[1], "%lf", &m);
    sscanf(argc[2], "%d", &R);
    read_data(argc[3], argc[4]);
    if (*argc[6] == 'c')
        init_memberships();
    else
        new_memberships(m);
    r = 0;
    while (r < R) {
        new_cluster_centers(m);
        e = new_memberships(m);
        r++;
        if (e < 0.001)
            break;
```

```
    }
    dump_stats();
    write_data(argc[5]);
    return 0;
}
```